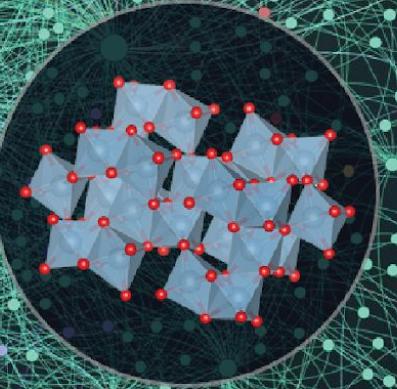
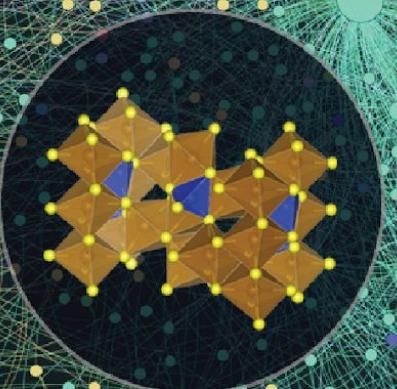


Crystal Structure graphs



Outline

Components, directionality, tasks

Molecules and crystals as graphs

graph connectivity and centralities

Matrix representation

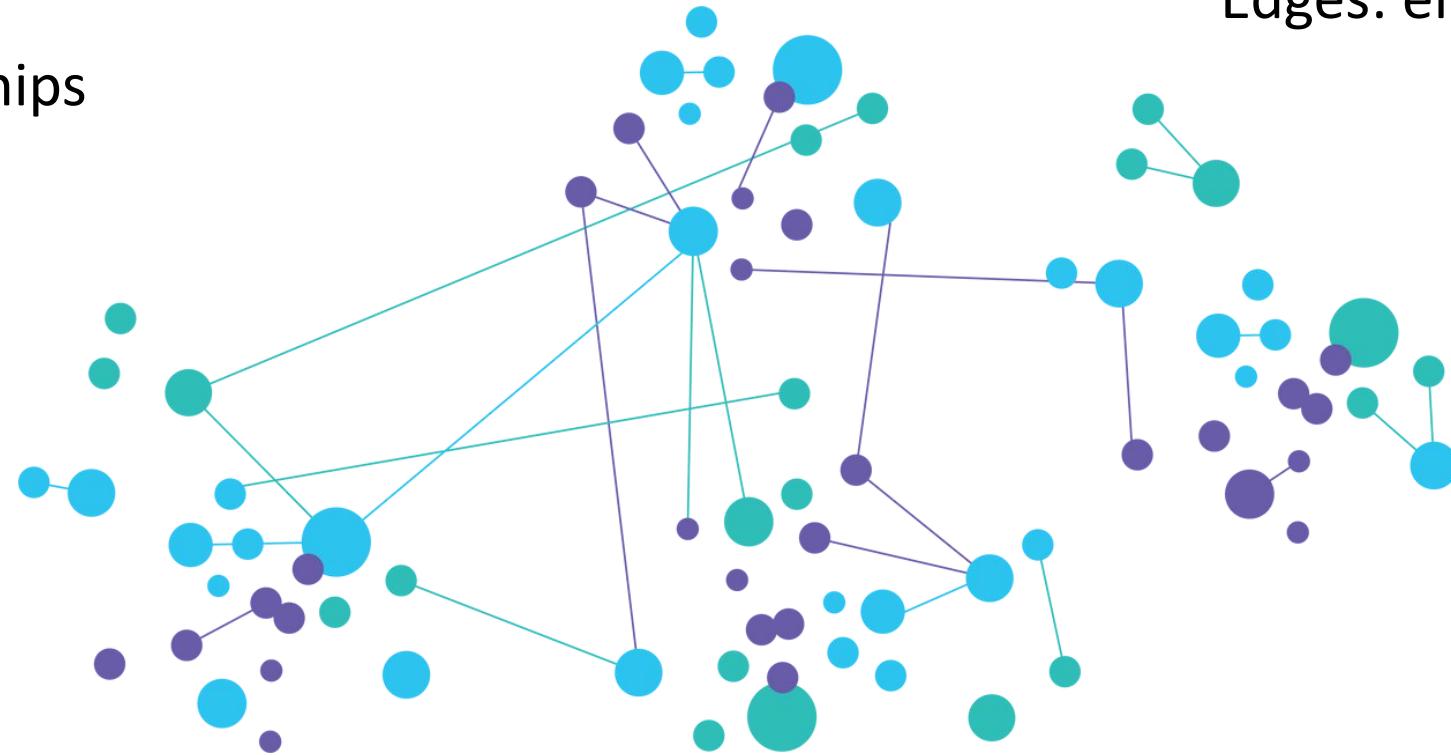
message passing and attention

Architecture design options

What should we look for in future graph network architectures?

Graphs are a convenient and powerful way to represent information

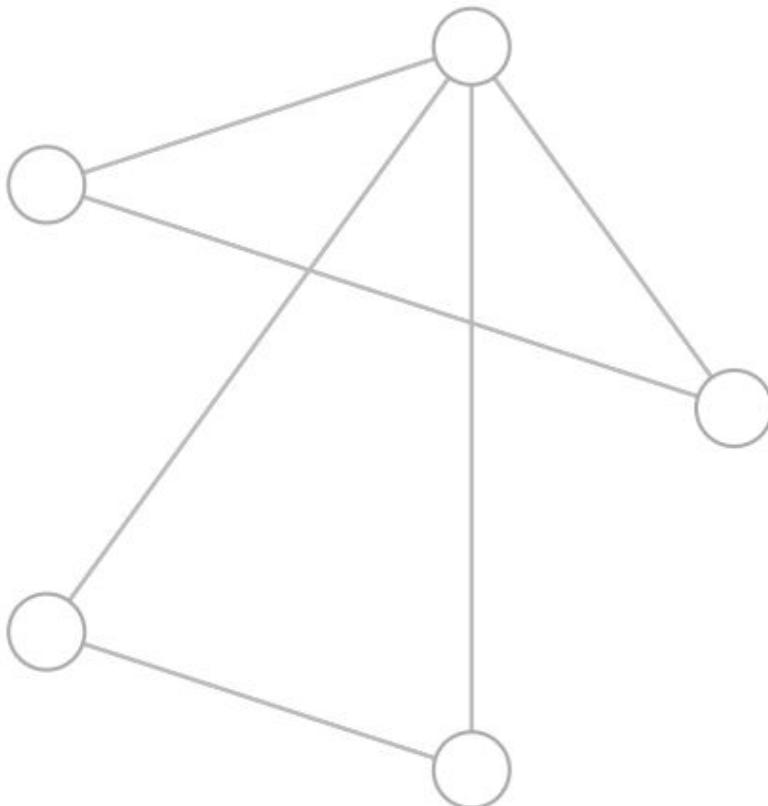
Nodes: people
Edges: relationships



Nodes: papers
Edges: citations

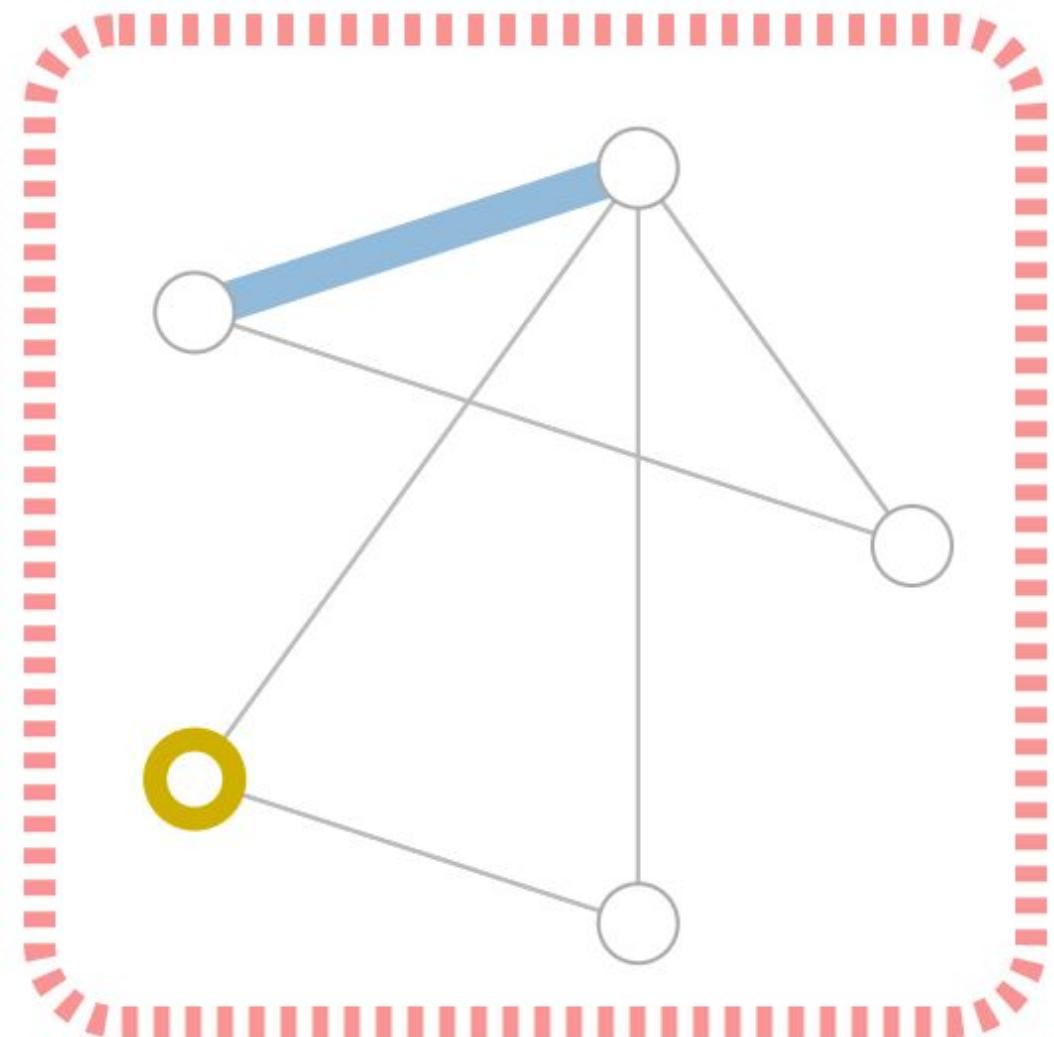
Nodes: email users
Edges: email message

Graphs contain key components



- V** Vertex (or node) attributes
e.g., node identity, number of neighbors
- E** Edge (or link) attributes and directions
e.g., edge identity, edge weight
- U** Global (or master node) attributes
e.g., number of nodes, longest path

We can encode information at all of these components



Vertex (or node) embedding



Edge (or link) attributes and embedding

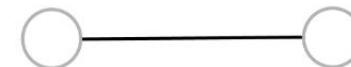


Global (or master node) embedding

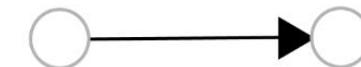


Another way of encoding information is in **direction** of edges

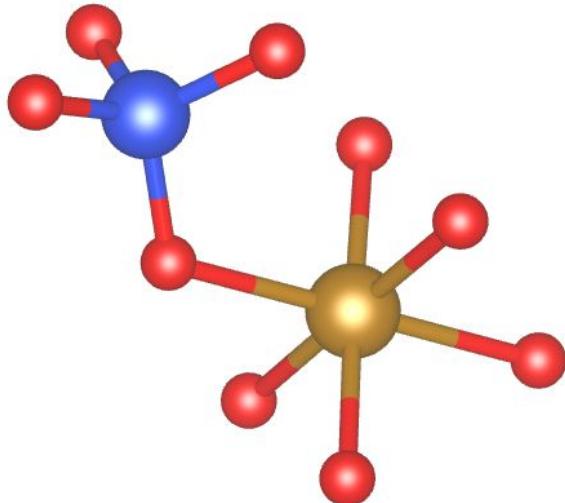
Undirected edge



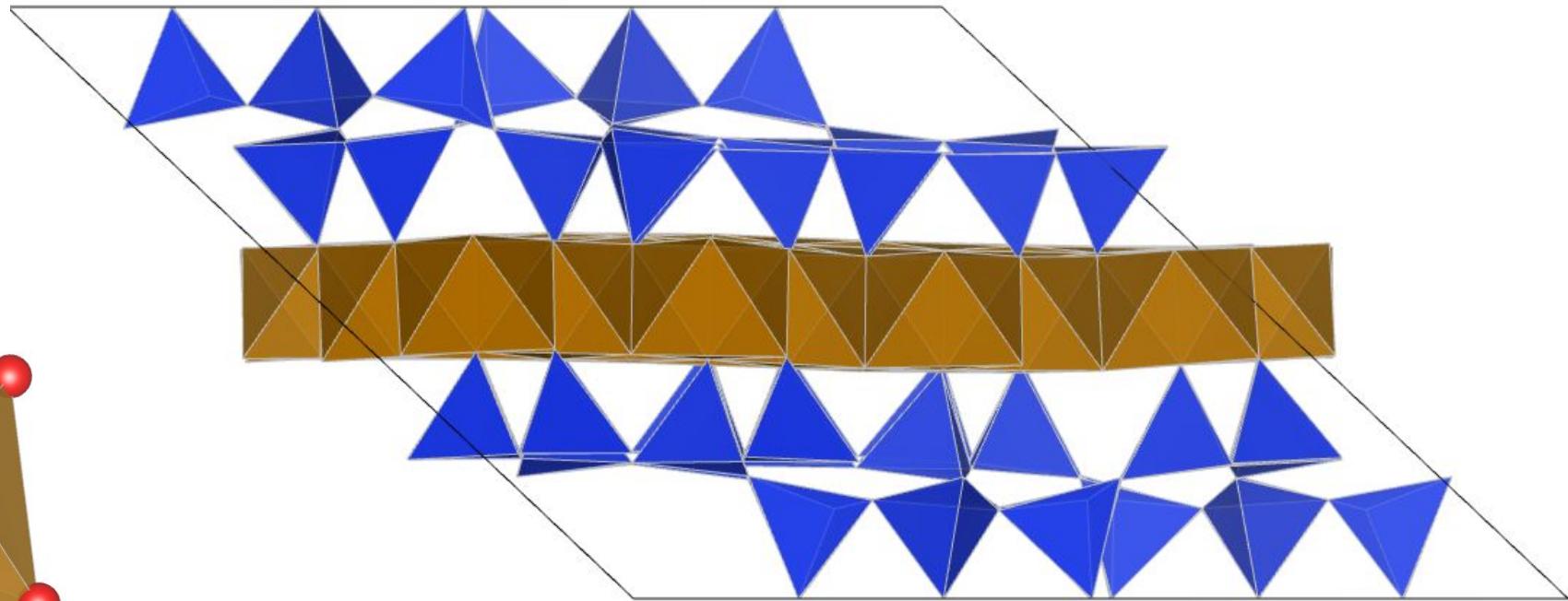
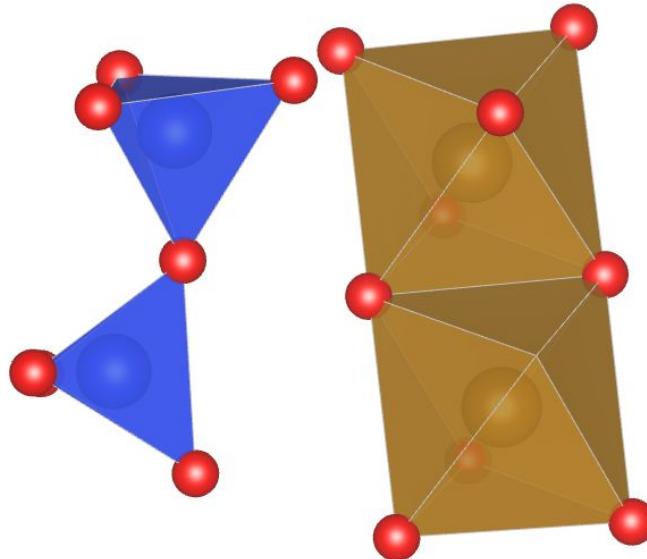
Directed edge



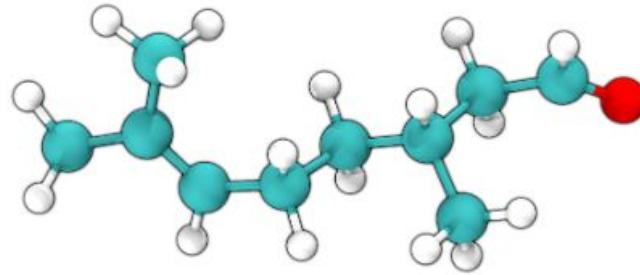
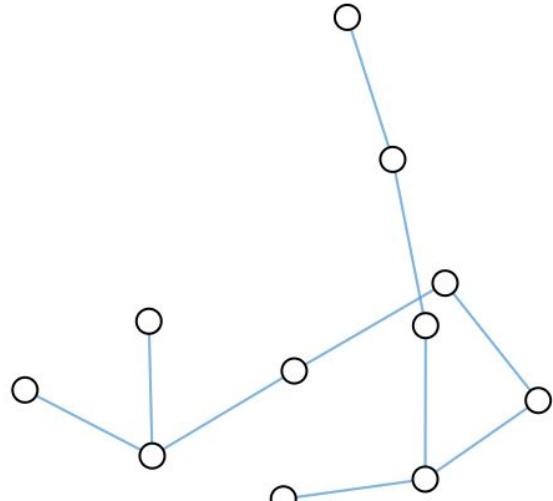
Molecules and crystals naturally lend themselves to graph networks



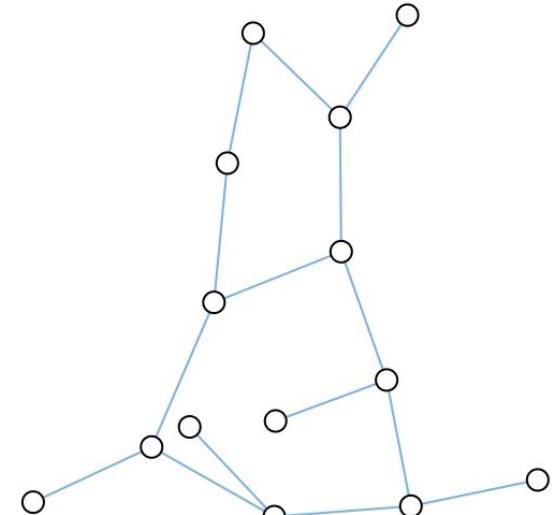
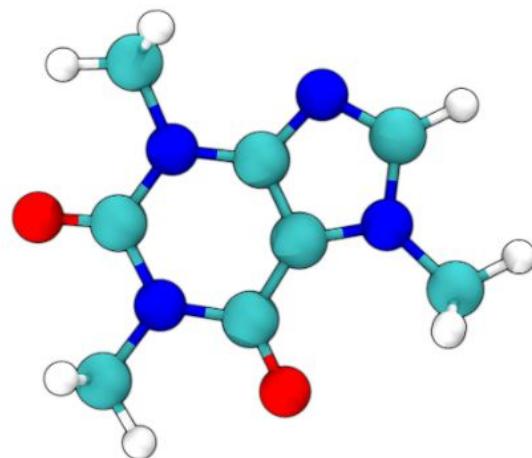
Nodes: atoms
Edges: bonds



Molecules look a bit different as graphs.



Why graphs belong to which?



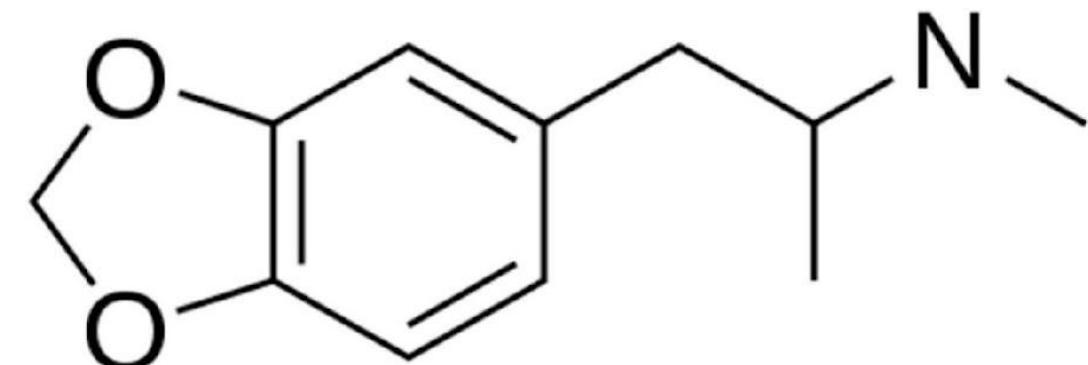
Strings are a common alternative representation

String-based Representation

SMILES: CC(CC1=CC2=C(C=C1)OCO2)NC

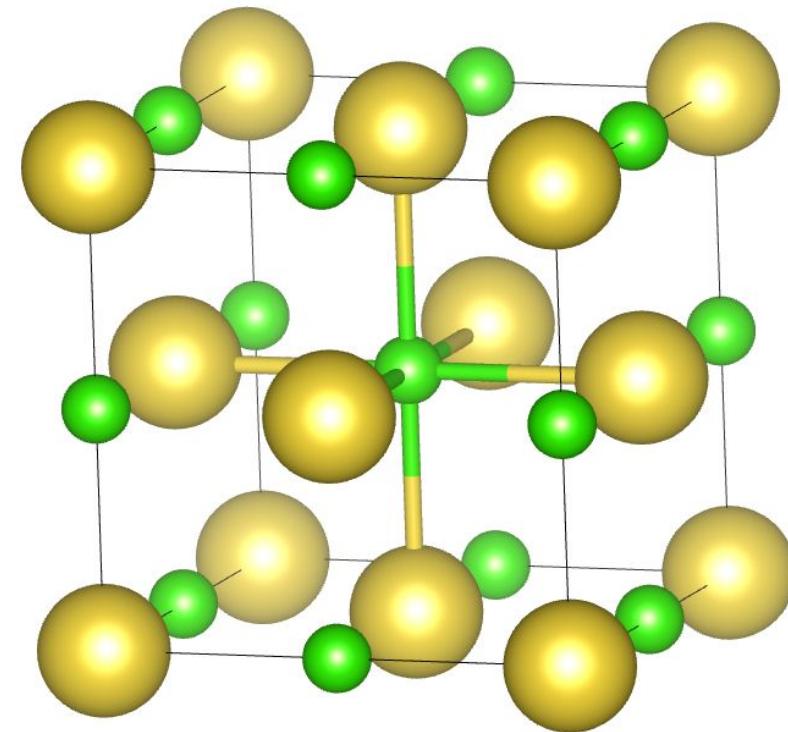
DeepSMILES: CCCCC=CC=CC=C6)) OCO5)))))))) NC

SELFIES: [C] [C] [B1_1] [P] [C] [C] [=C] [C] [=C]
[B1_1] [B1_1] [C] [=C] [R1] [B1_2]
[O] [C] [O] [R1] [B1_3] [N] [C]



Graphs for periodic lattices are more challenging!

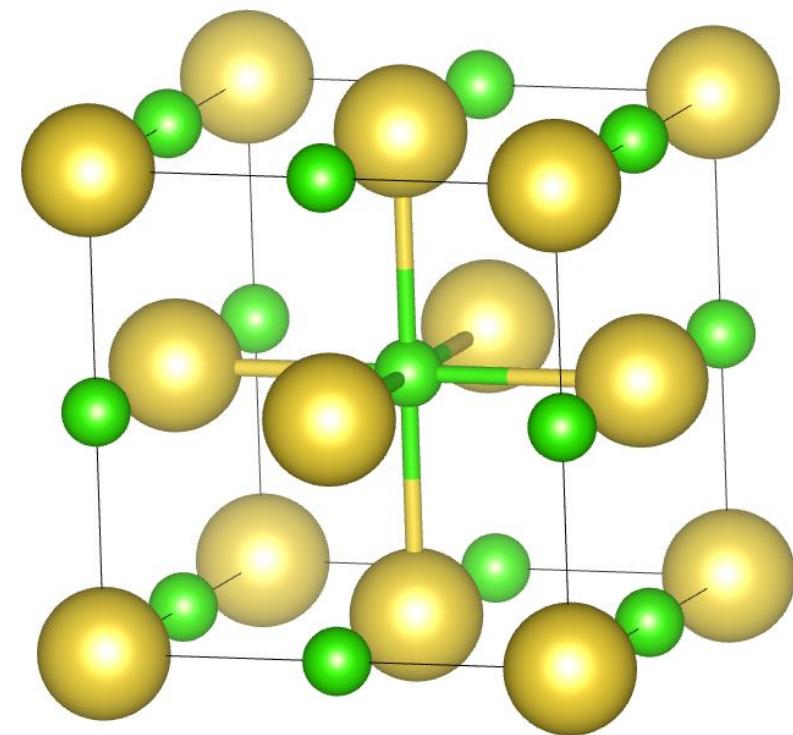
$$2\text{\AA} < x < 3\text{\AA}$$



6 neighbors

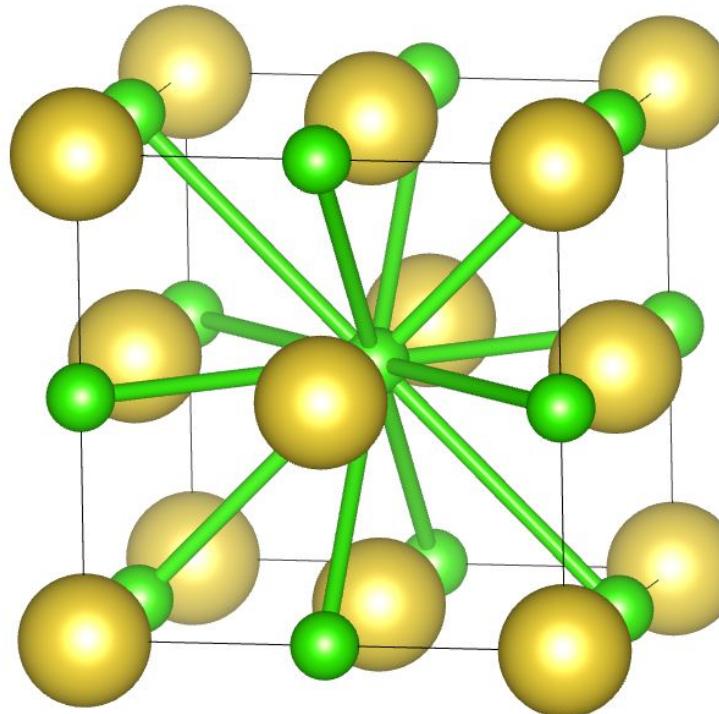
Graphs for periodic lattices are more challenging!

$2\text{\AA} < x < 3\text{\AA}$



6 neighbors

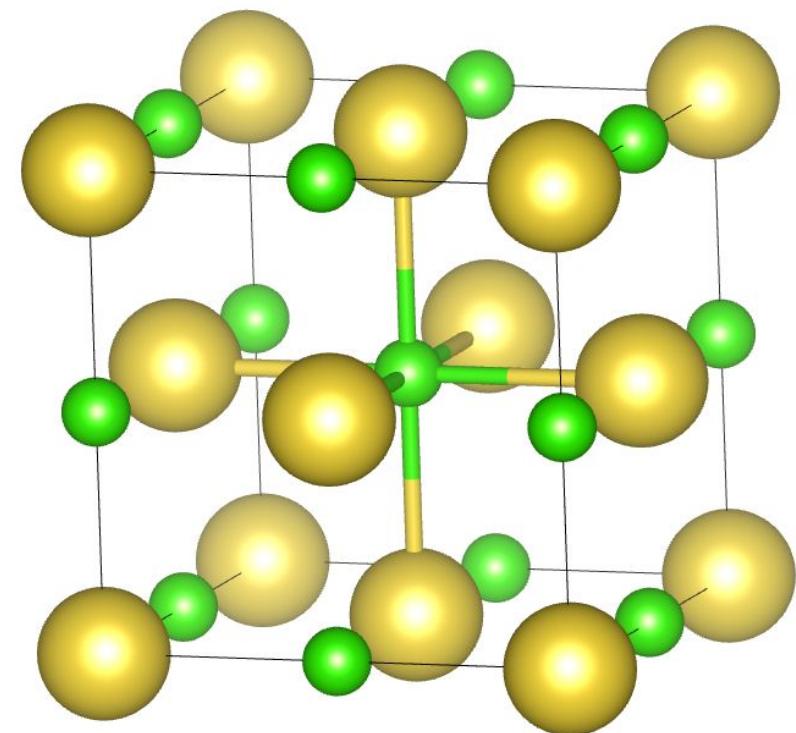
$3\text{\AA} < x < 4\text{\AA}$



12 neighbors

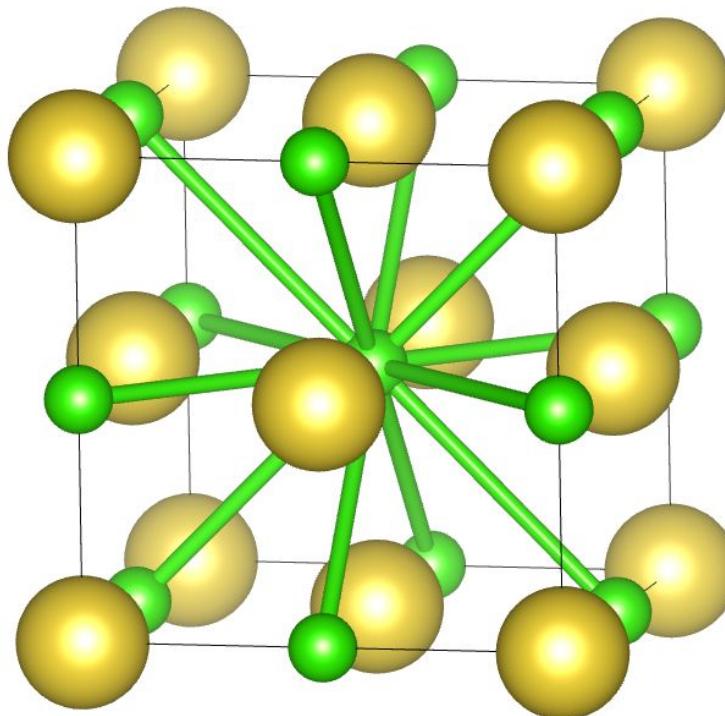
Graphs for periodic lattices are more challenging!

$2\text{\AA} < x < 3\text{\AA}$



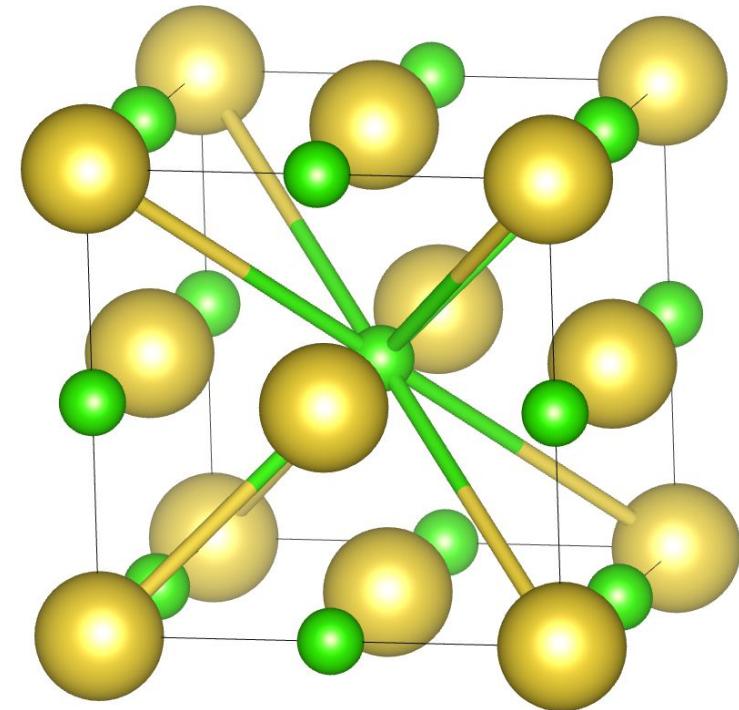
6 neighbors

$3\text{\AA} < x < 4\text{\AA}$



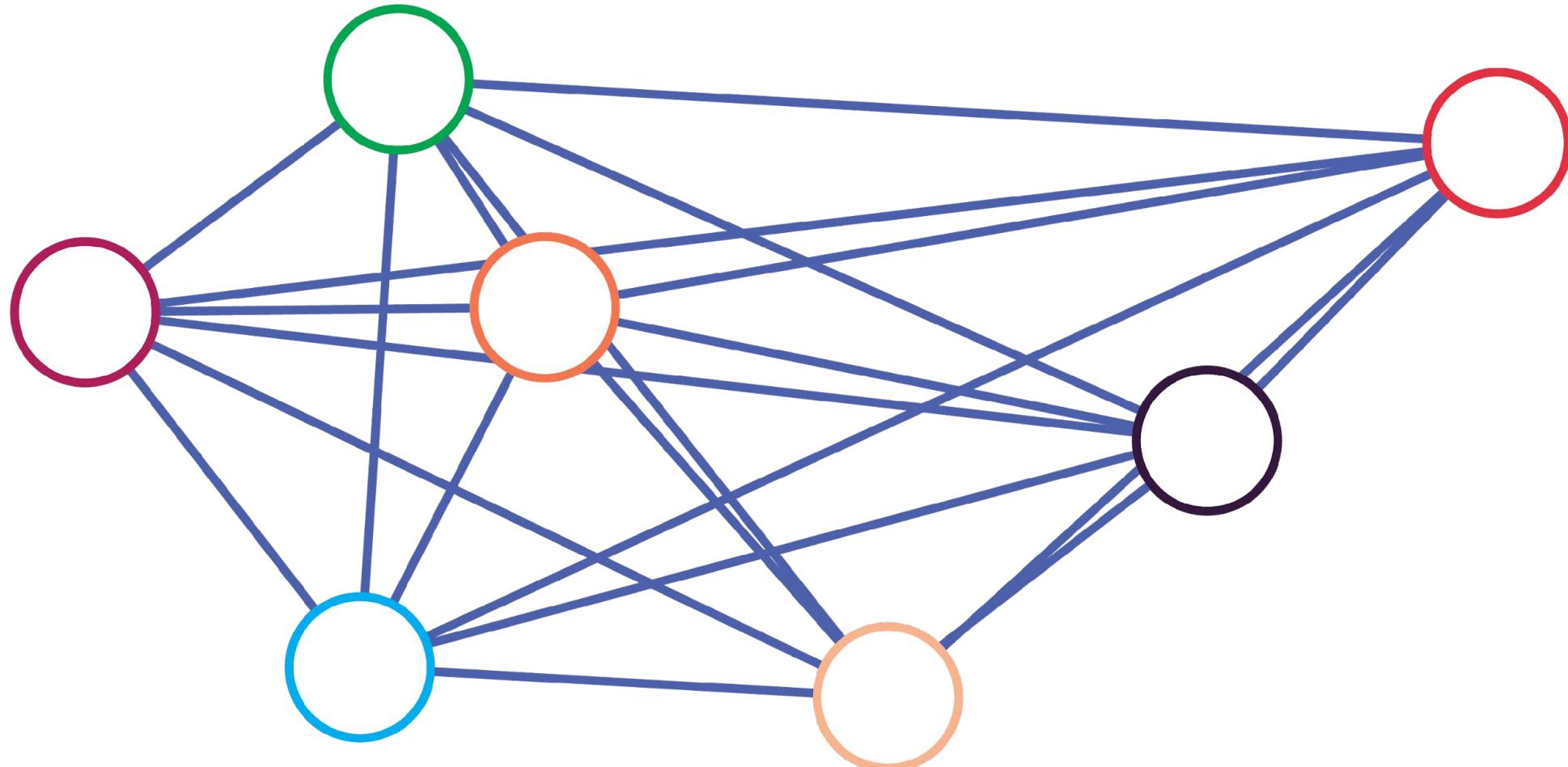
12 neighbors

$4\text{\AA} < x < 5\text{\AA}$

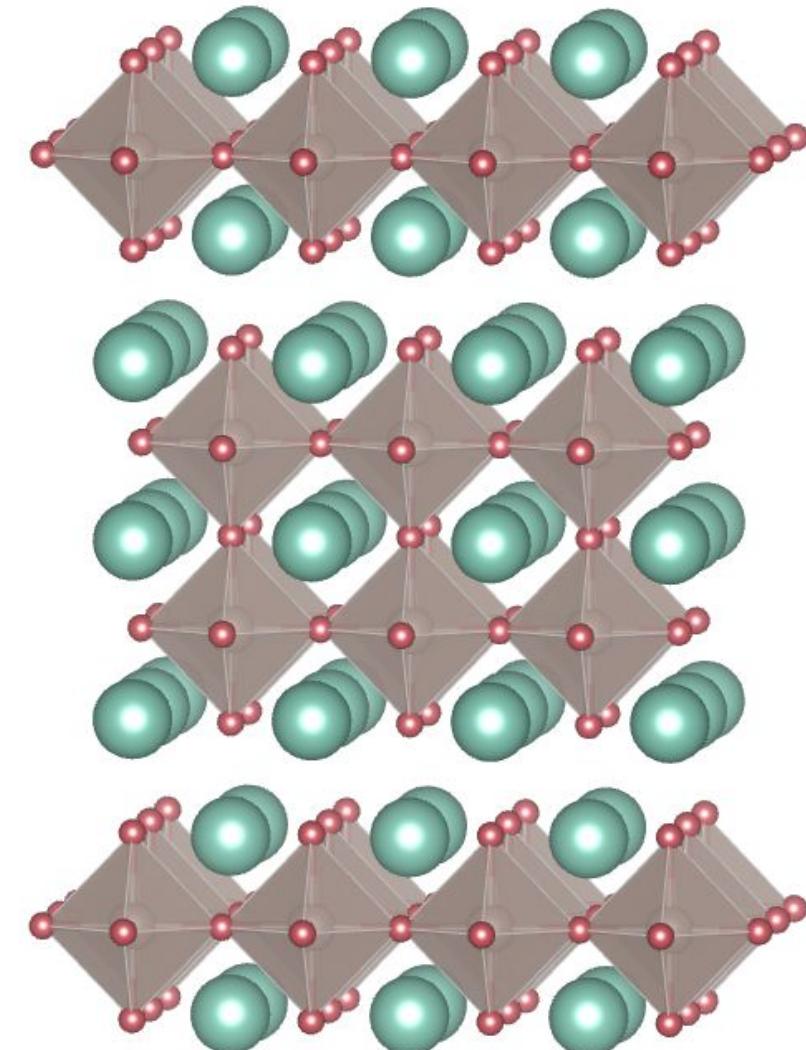
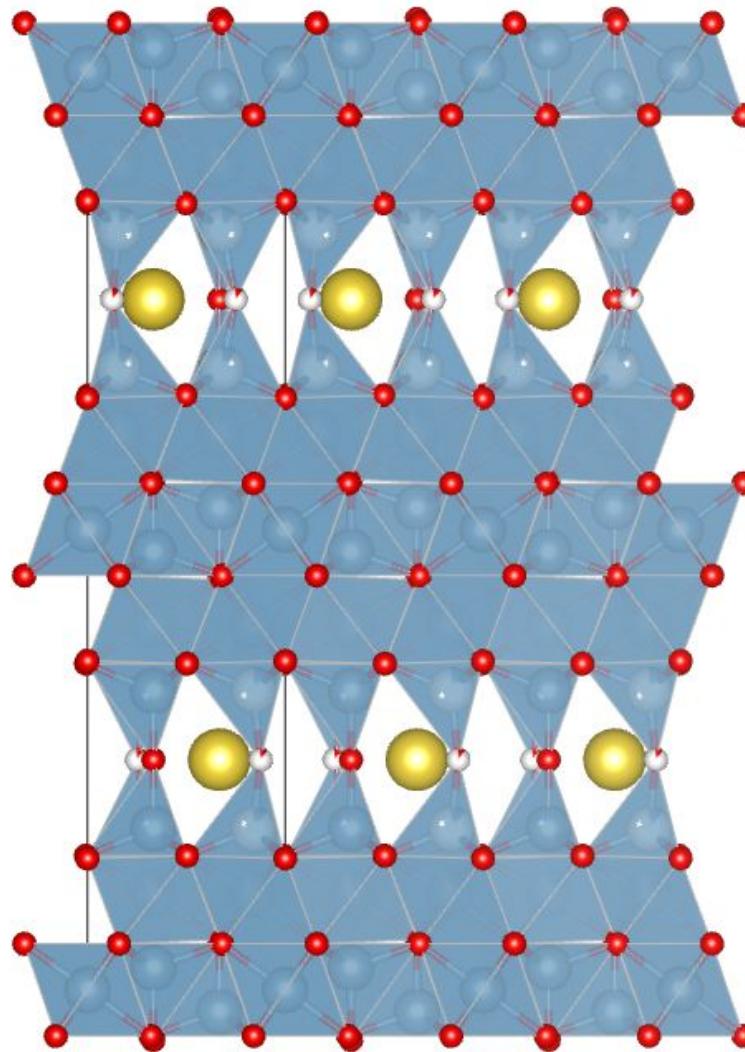
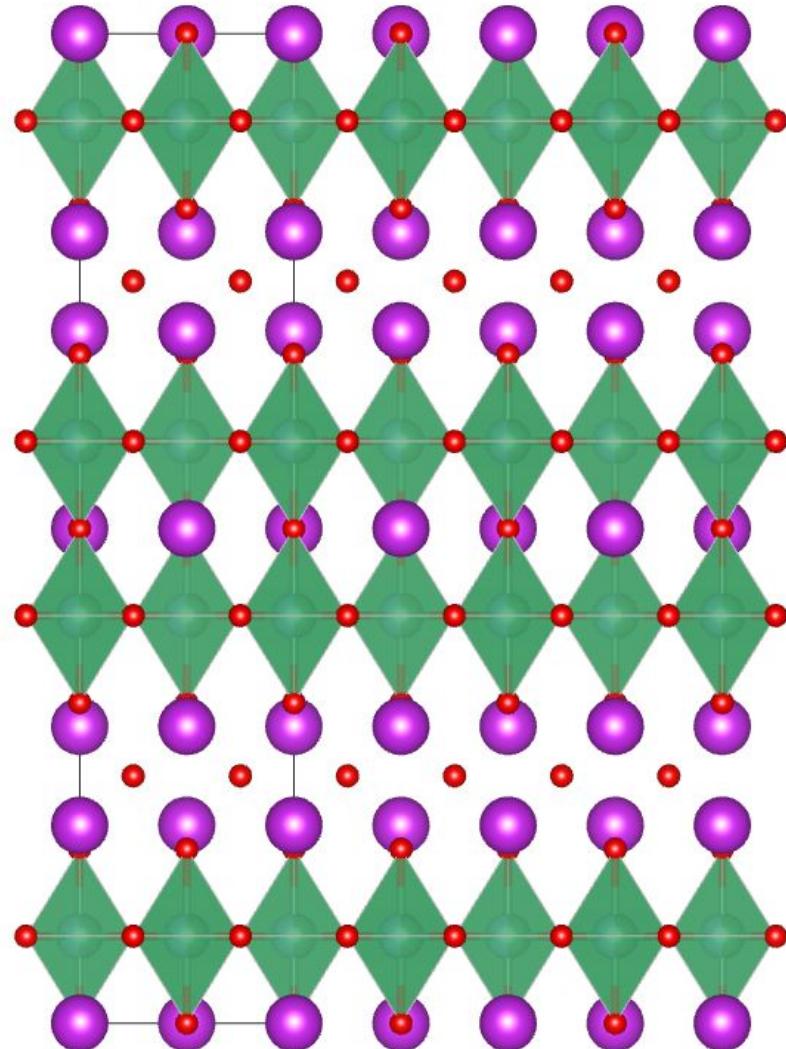


8 neighbors

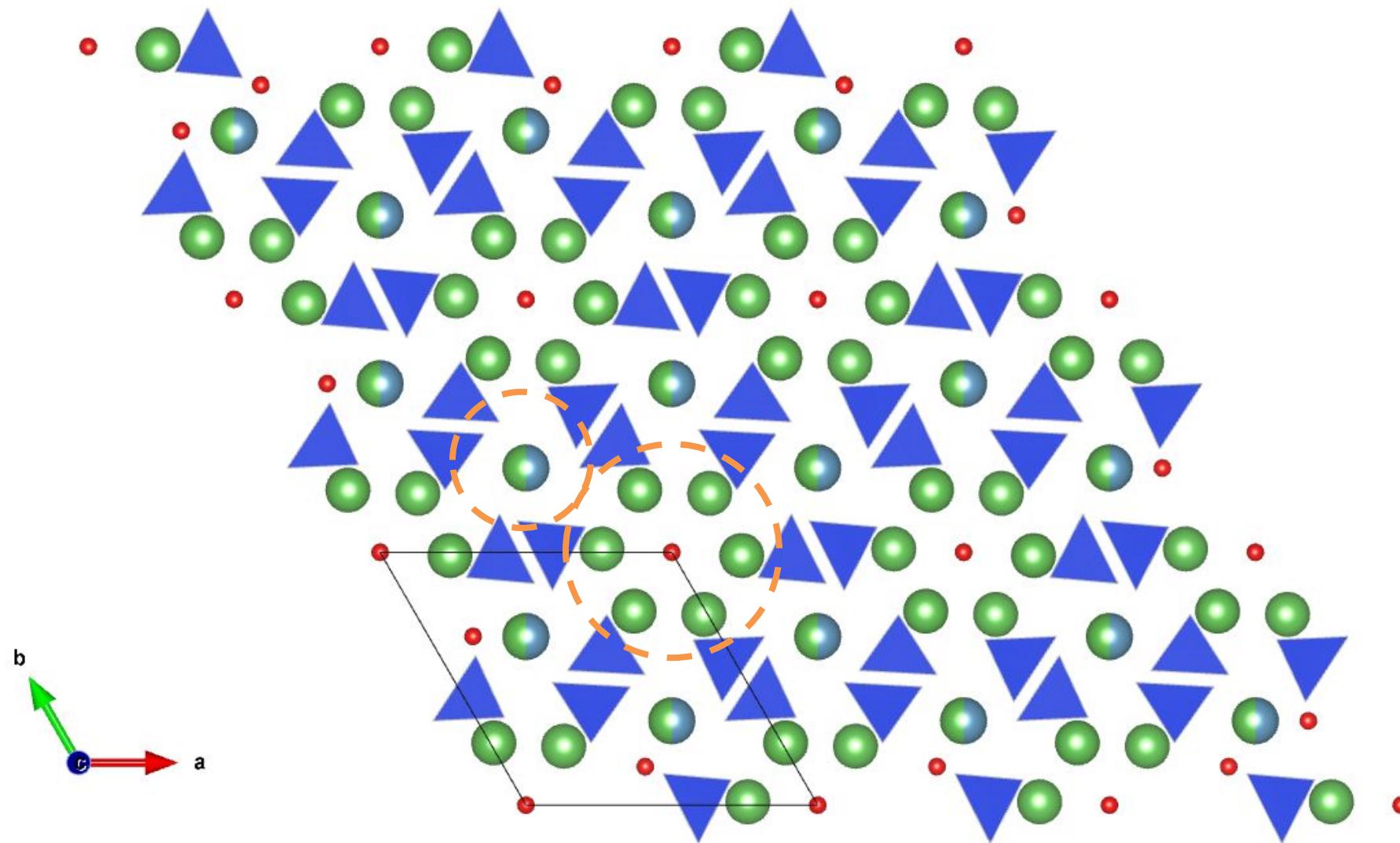
Current graph networks are fully connected



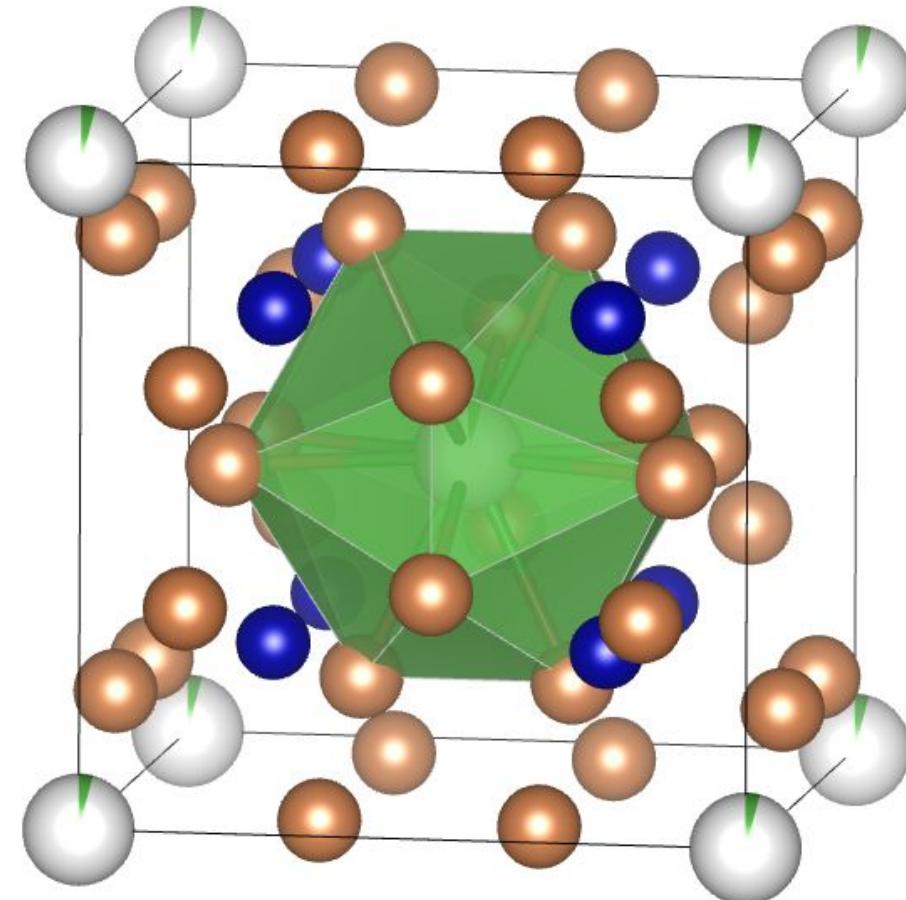
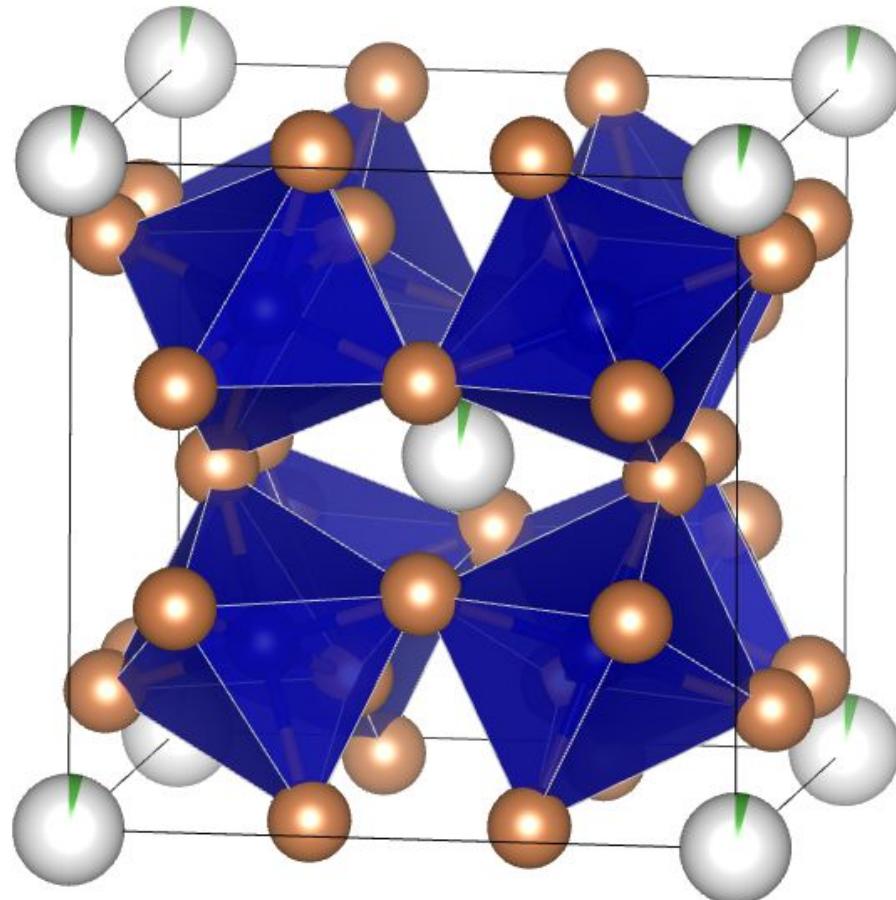
Many properties rely on specific interactions between atoms



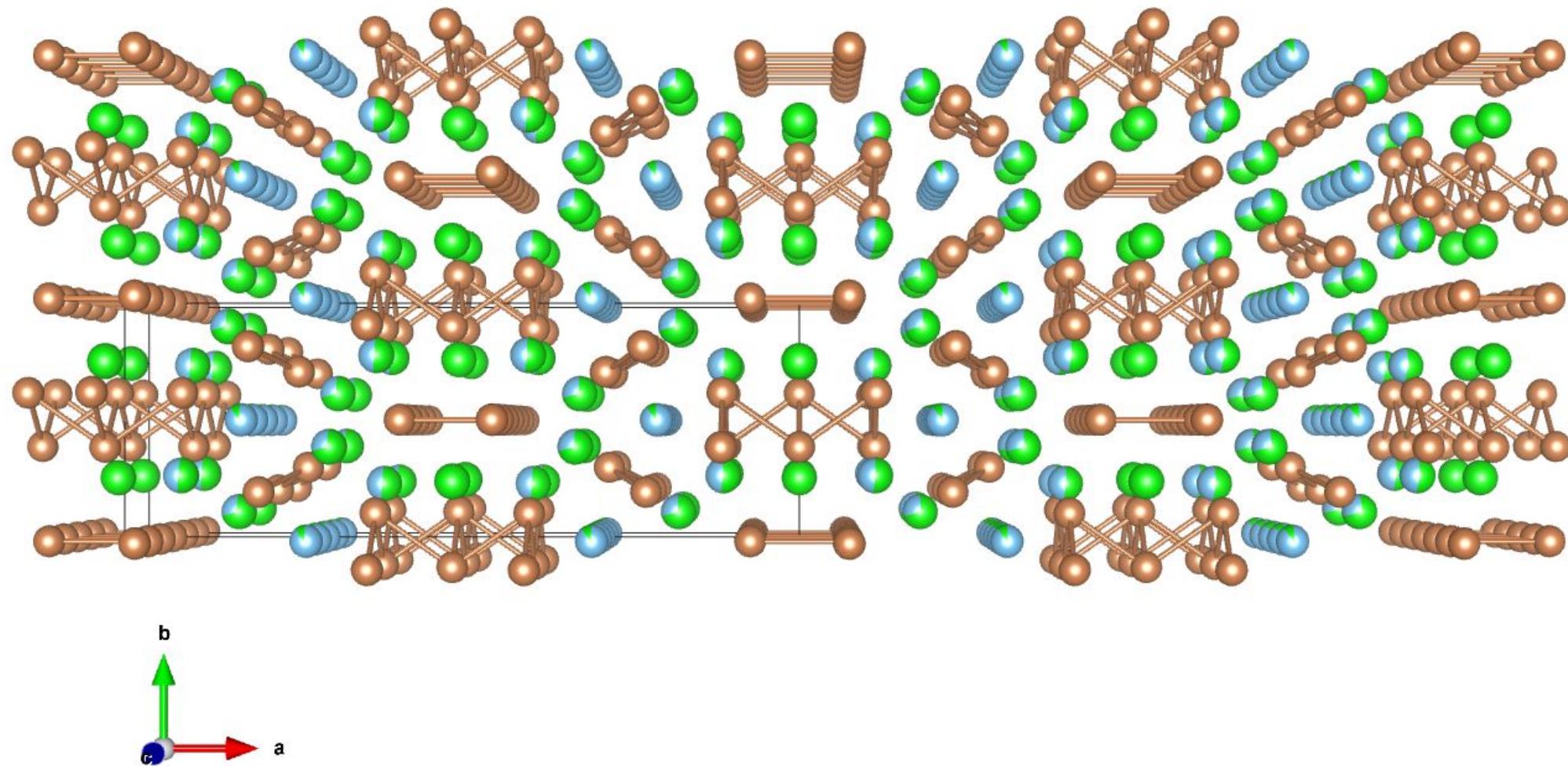
Many properties rely on specific interactions between atoms



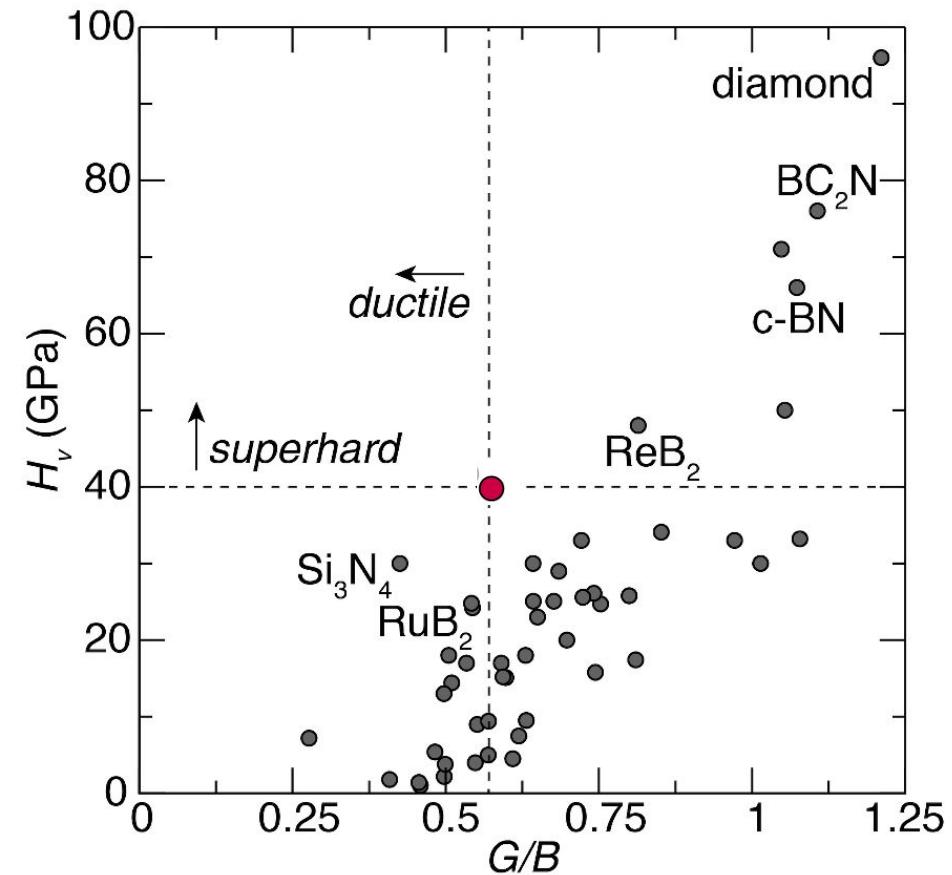
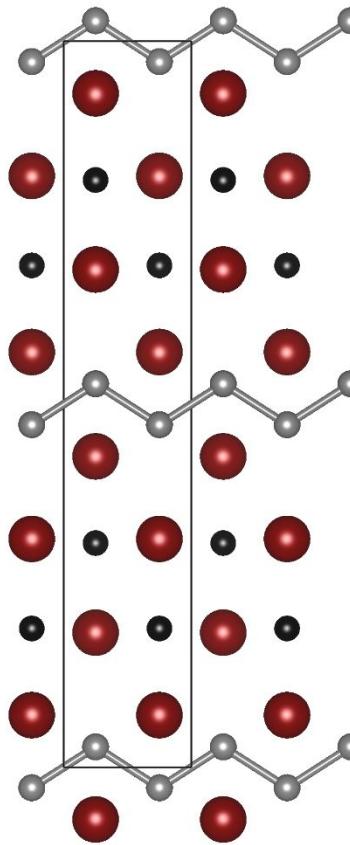
Many properties rely on specific interactions between atoms



Many properties rely on specific interactions between atoms



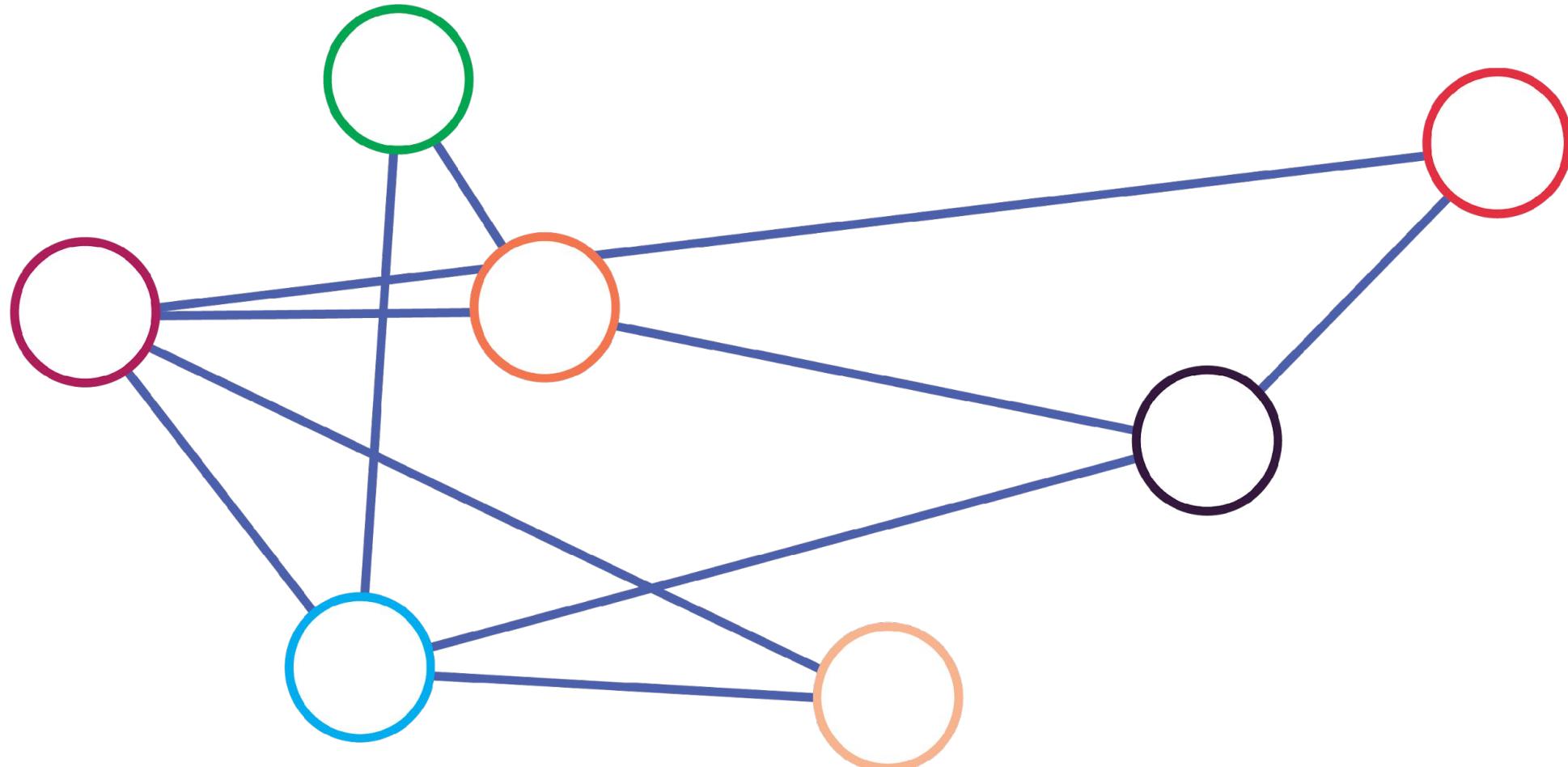
Many properties rely on specific interactions between atoms



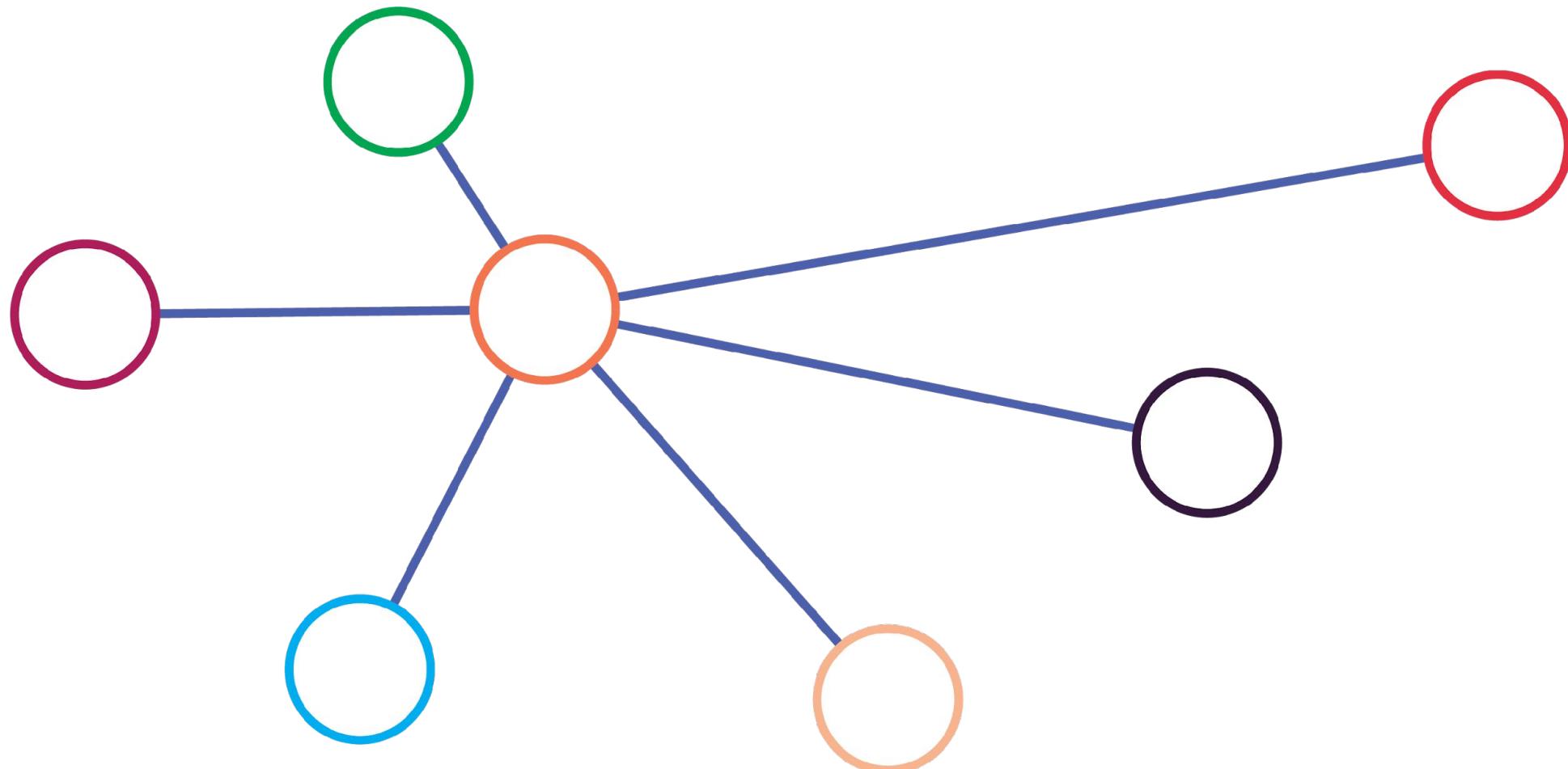
Mansouri et al. (2019) Chem. Mater.

Mansouri et al. (2018) JACS, Parry et al. (2019) J Mat Chem A

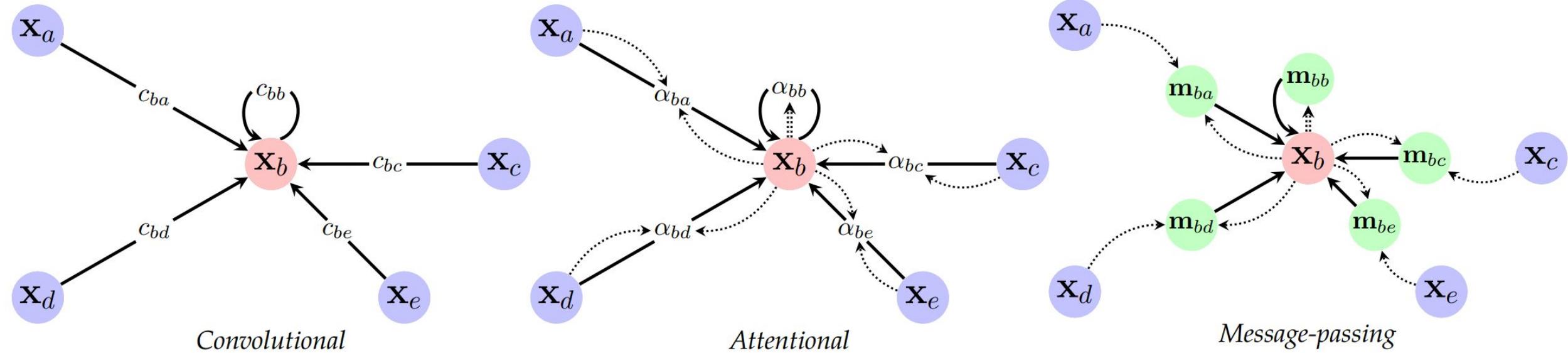
Sparse graphs offer unique nodal information and faster
computation



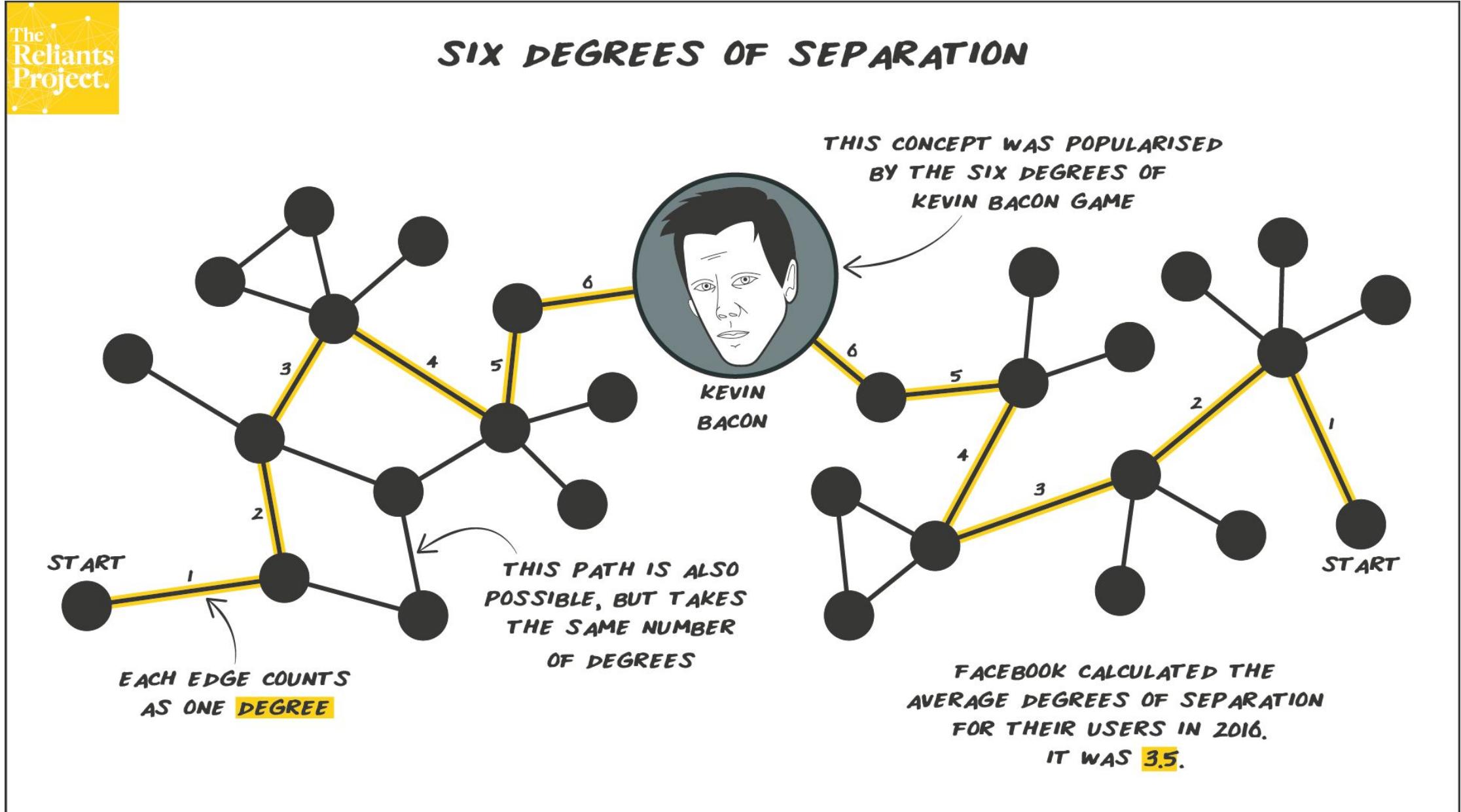
Connectivity impacts aggregation, attention, and message passing!



Connectivity impacts aggregation, attention, and message passing!



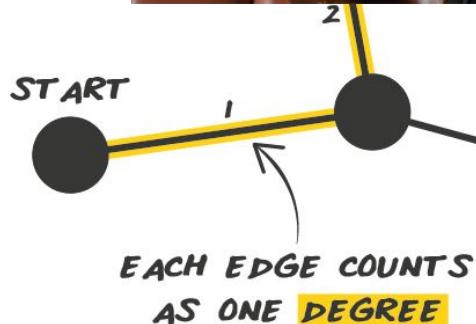
Many centralities can be considered for a given network



Many centralities can be considered for a given network



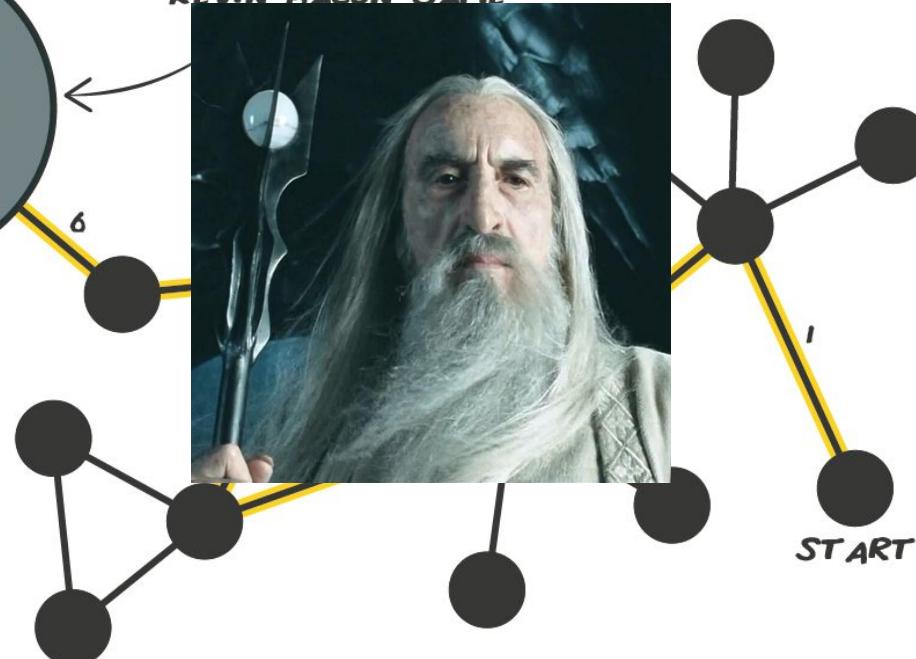
SIX DEGREES OF SEPARATION



THIS PATH IS ALSO POSSIBLE, BUT TAKES THE SAME NUMBER OF DEGREES

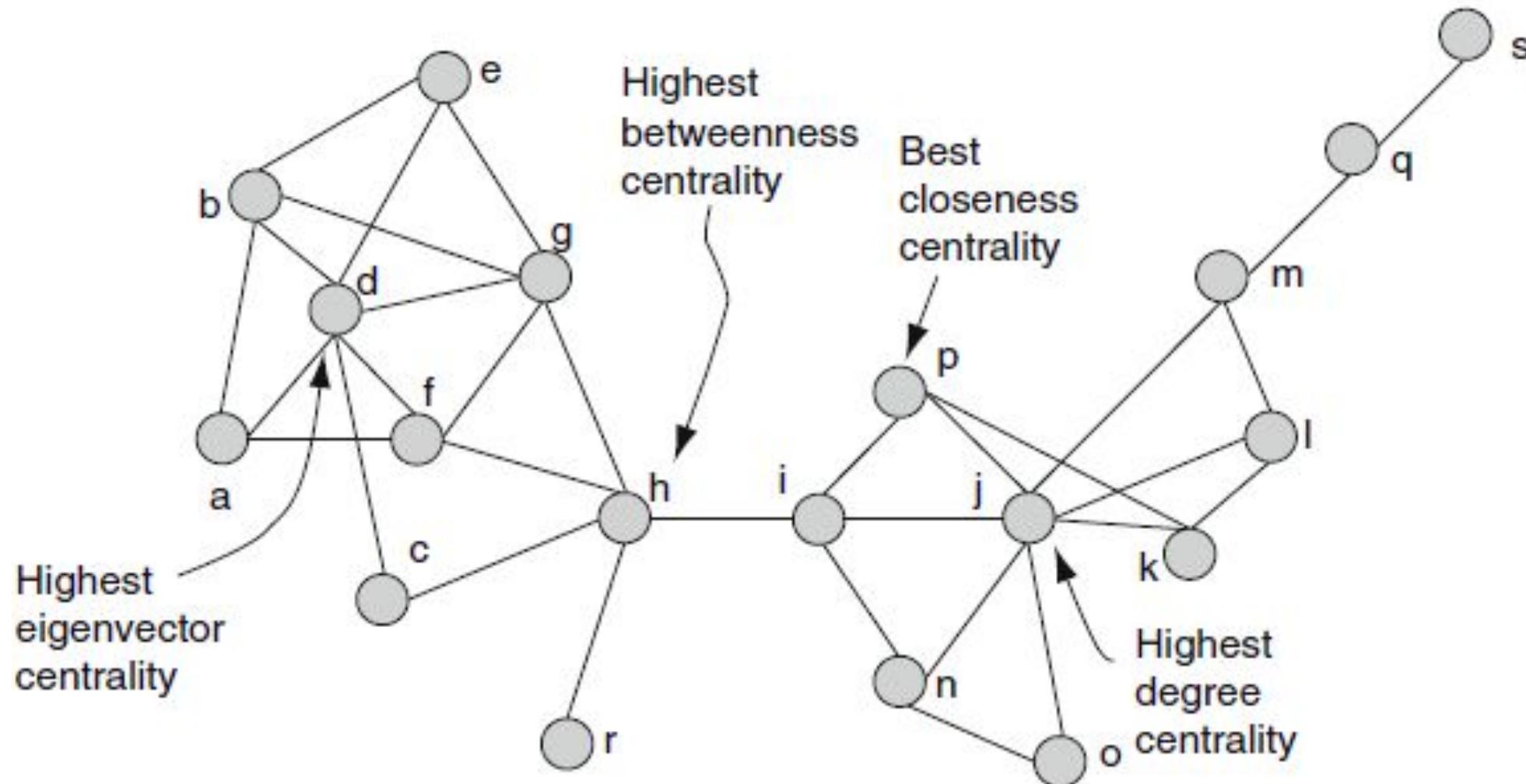


THIS CONCEPT WAS POPULARISED BY THE SIX DEGREES OF KEVIN BACON GAME

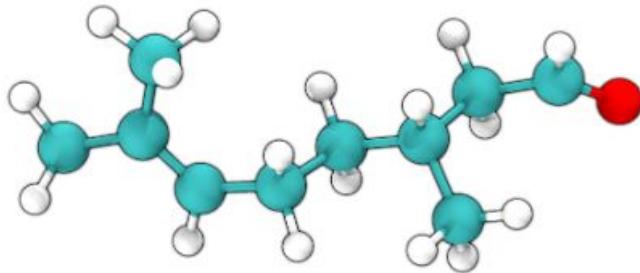
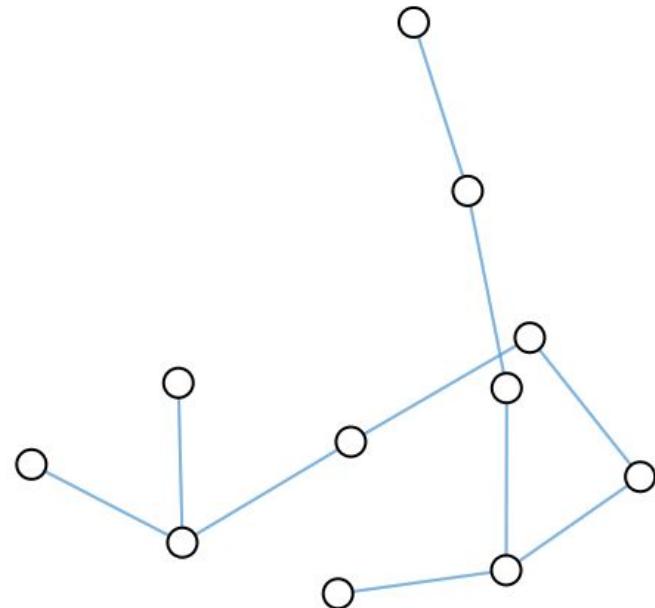


FACEBOOK CALCULATED THE AVERAGE DEGREES OF SEPARATION FOR THEIR USERS IN 2016.
IT WAS 3.5.

Many centralities can be considered for a given network



Tasks can be broken down to node, edge, and graph level

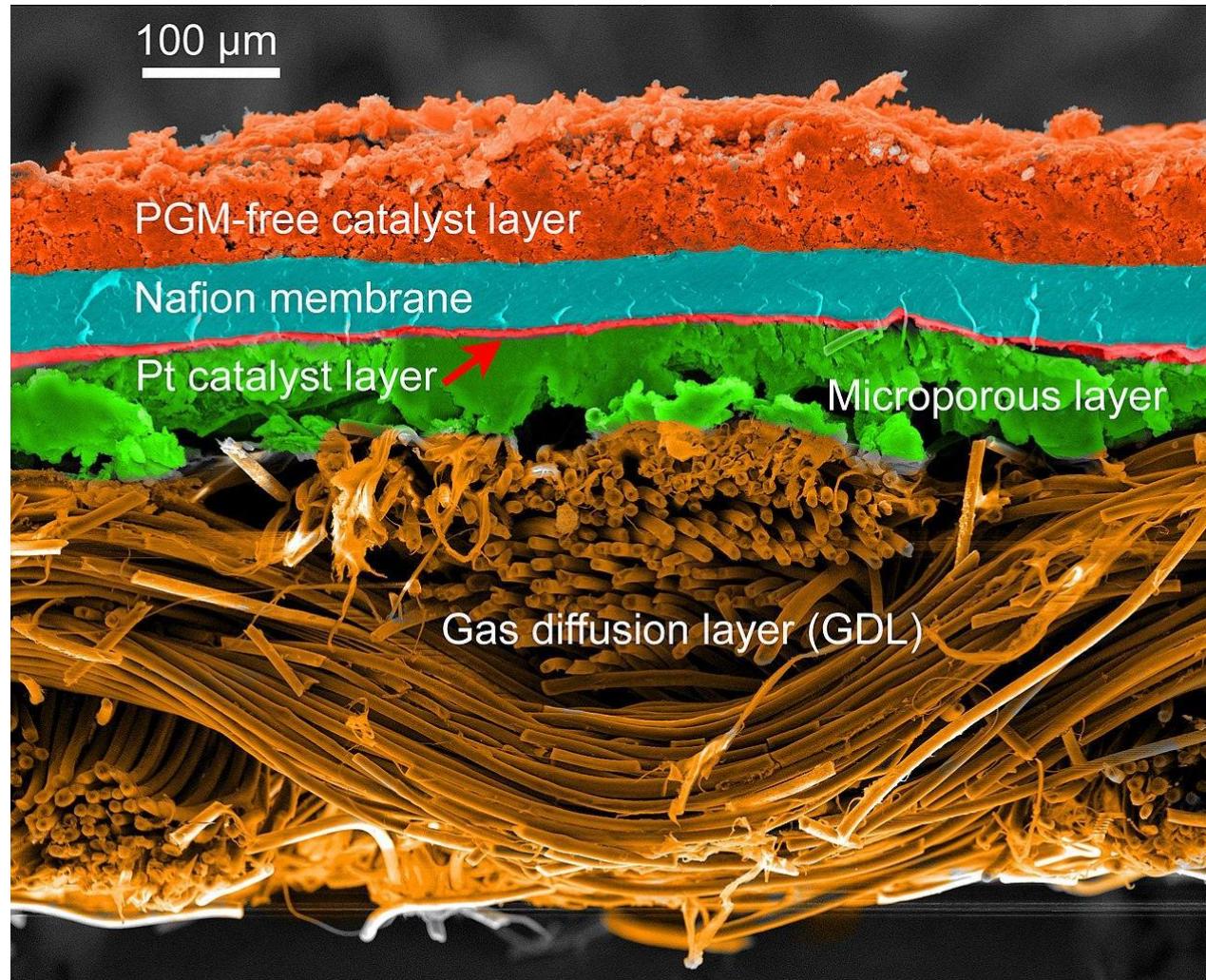


Graph level?

Node level?

Edge level?

Images can be thought of as graphs too!



https://commons.wikimedia.org/wiki/File:SEM_micrograph_of_an_MEA_cross_section.jpg

Neural networks require rectangular or grid like arrays of data inputs

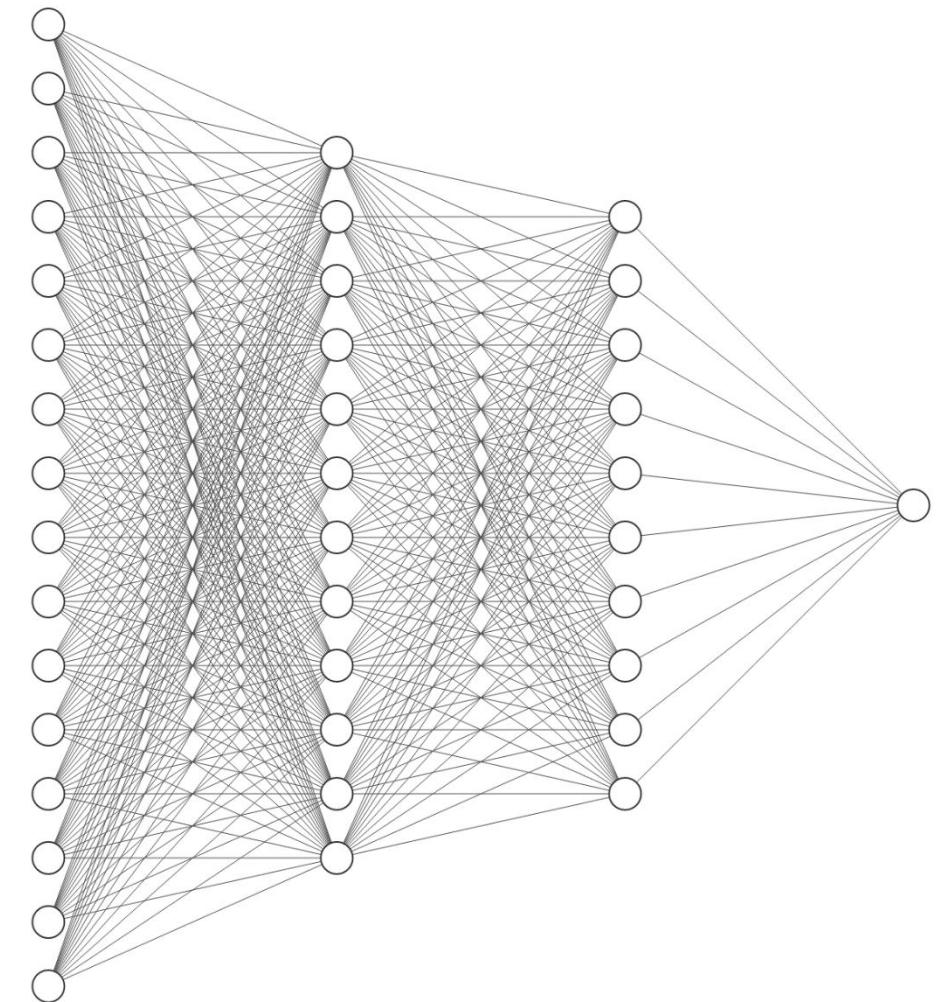
Nodes store numbers

- “activated” with a value between 0 and 1
- Functions that pass information

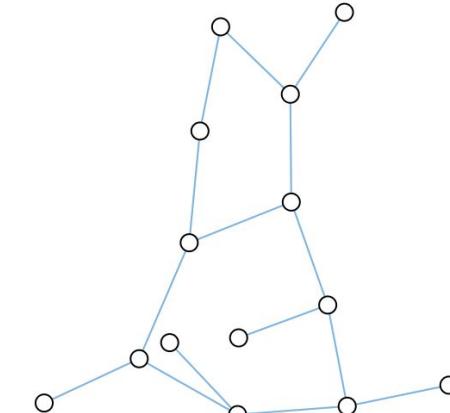
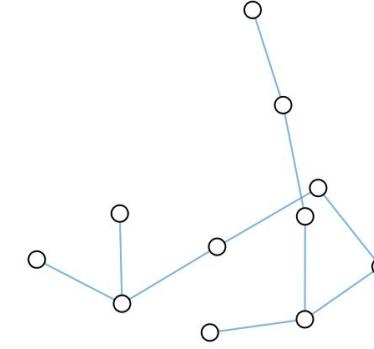
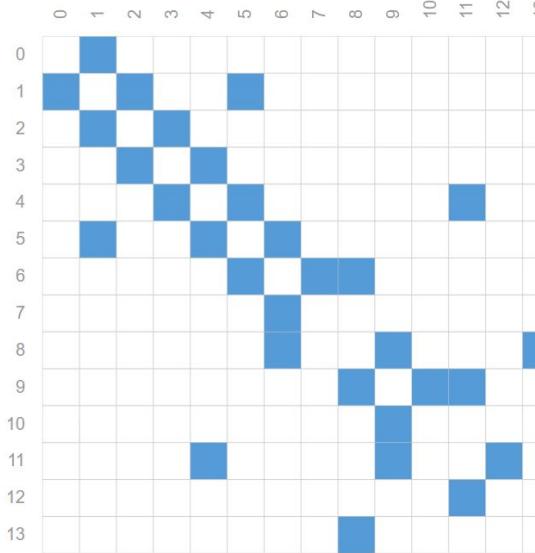
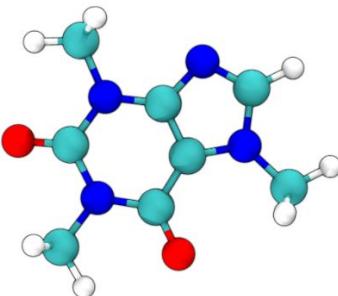
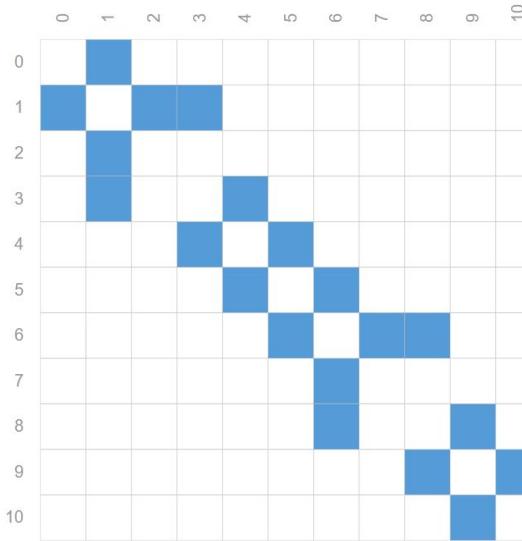
Result of network is a prediction (regression or classification)

Intermediate layers allow us to build up complex non-linearity

How do we make a graph compatible with NN?



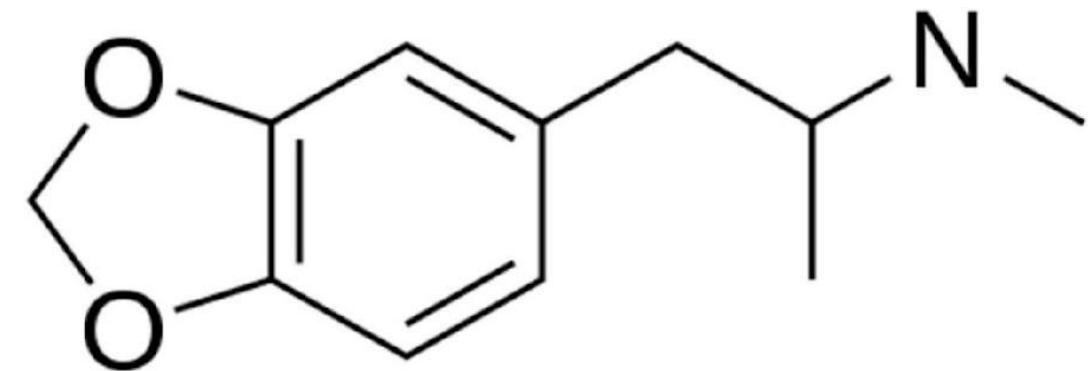
Adjacency matrices are one way to make graphs NN compatible



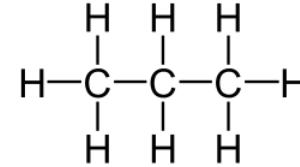
Matrices can contain different types of information

Adjacency-Matrix-based Representation

$$\left(\begin{array}{c|cccccccccccc} C & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ O & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ C & 0 & 1 & 0 & 0 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ O & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ C & 0 & 0 & 2 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ C & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\ C & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ C & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 1 & 0 & 0 \\ C & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 1 & 0 & 1 & 0 \\ C & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ N & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ C & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ C & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{array} \right)$$



Adjacency matrices have some limitations



Permutation invariance

- If you ordered points differently for the same structure, you should get the same representation
- Adjacency lists help here!

	c1	c2	c3			c1	c2	c3
c1	0	1	0			0	0	1
c2	1	0	1			0	0	1
c3	0	1	0			1	1	0

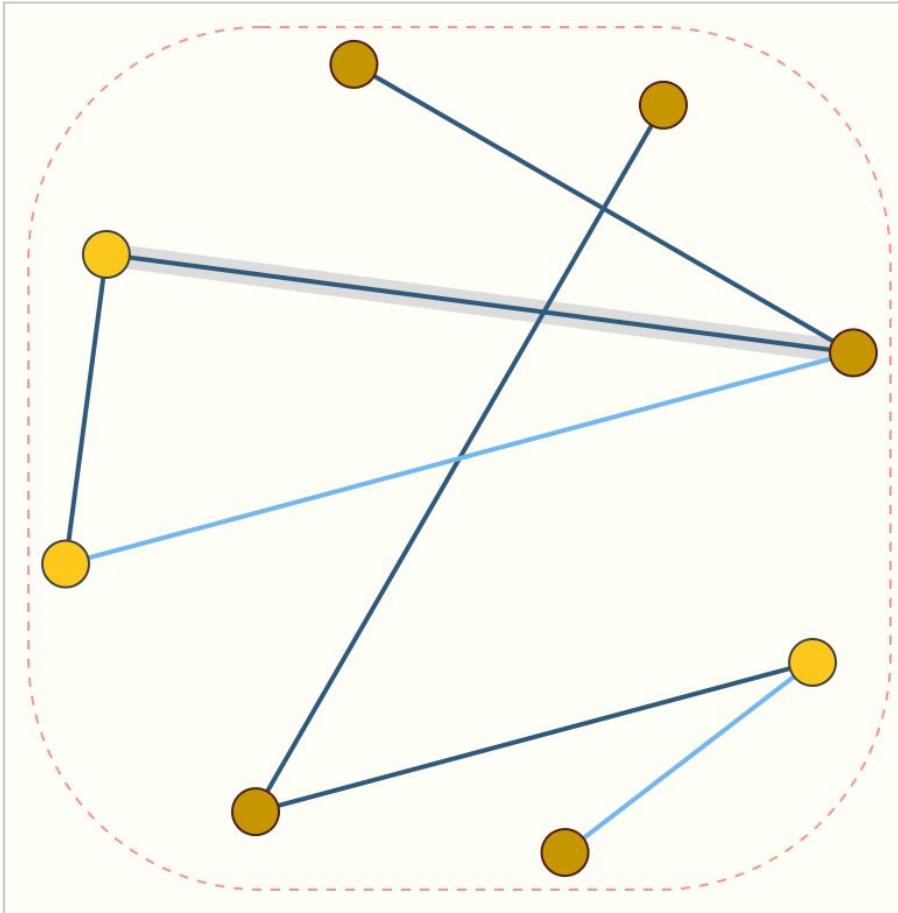
Adjacency-Matrix-based Representation

$$\begin{pmatrix} \text{C} \\ \text{O} \\ \text{C} \\ \text{O} \\ \text{C} \\ \text{C} \\ \text{C} \\ \text{C} \\ \text{C} \\ \text{C} \\ \text{N} \\ \text{C} \\ \text{C} \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

Representation should be space efficient

- Adjacency matrices can end up being very sparse if most nodes are only connected to a few others of many

Remember, graphs have 4 types of info!



Nodes

[0 , 1 , 1 , 0 , 0 , 1 , 1 , 1]

Edges

[2 , 1 , 1 , 1 , 2 , 1 , 1]

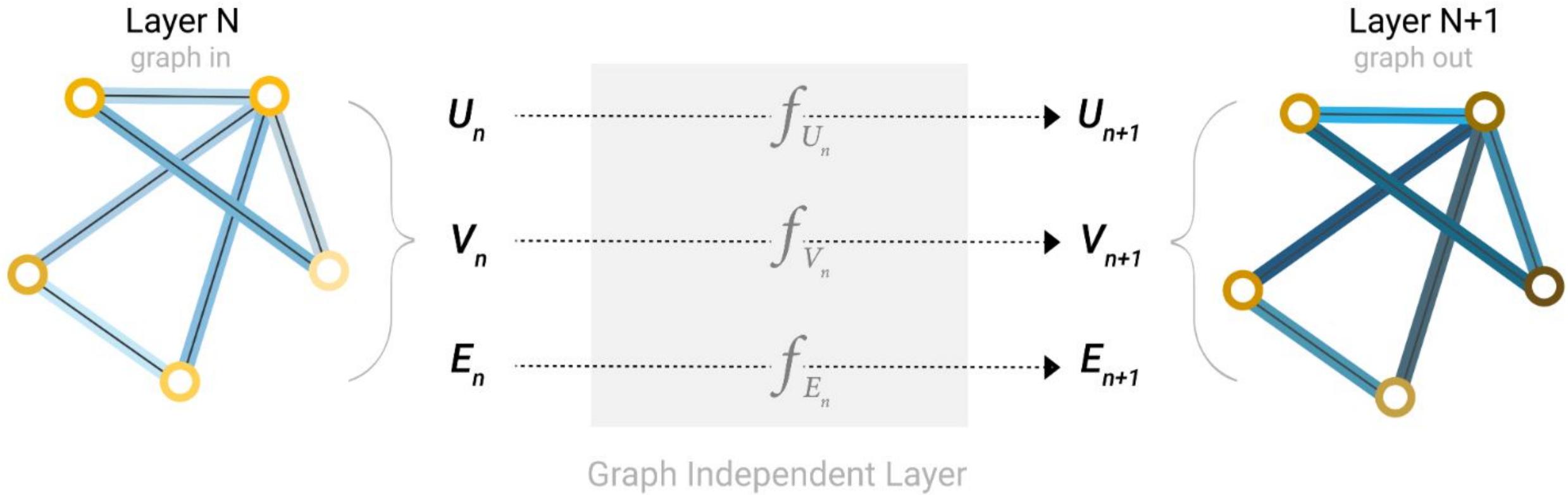
Adjacency List

[[1, 0] , [2, 0] , [4, 3] , [6, 2] ,
[7, 3] , [7, 4] , [7, 5]]

Global

0

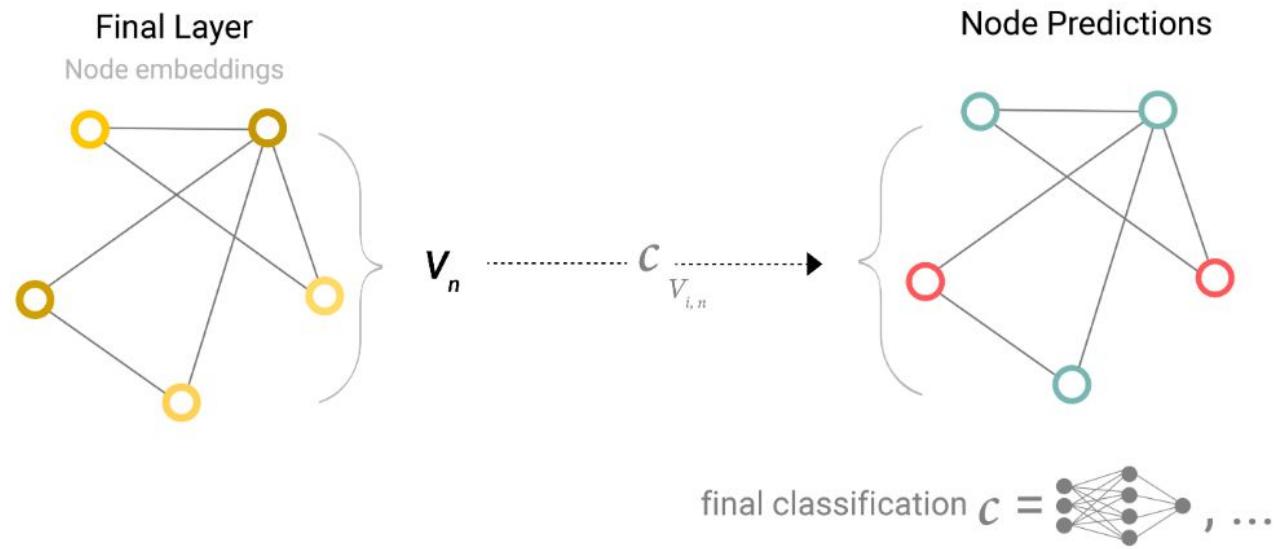
Most GNNs use graph in - graph out approach



Simplest would be to use MLP for each node, edge, and graph attribute.

update function $f = \begin{array}{c} \text{MLP} \\ \vdots \end{array}, \dots$

We have various options when it comes to making predictions

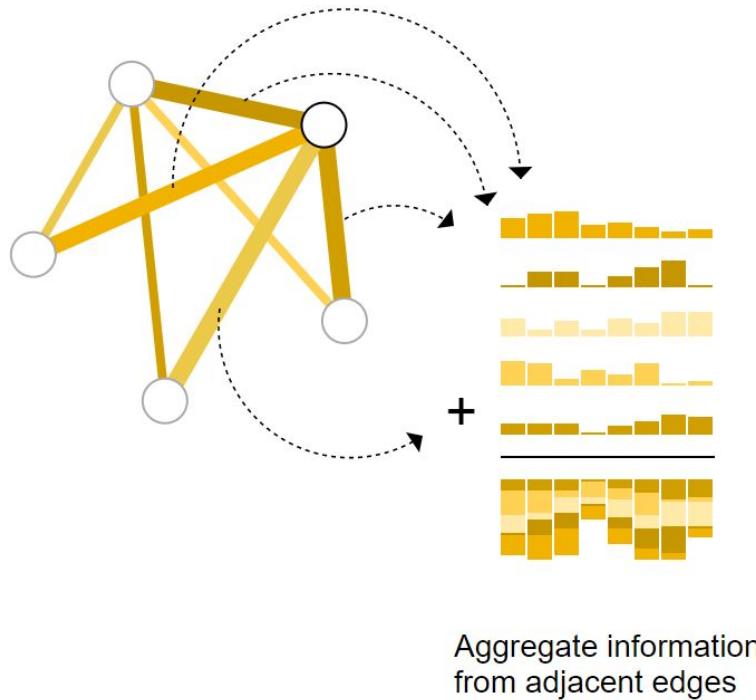


If you have information for each node and you need to make a prediction for each node

- Simply apply an activation function!

What if node prediction needed with only edge info?

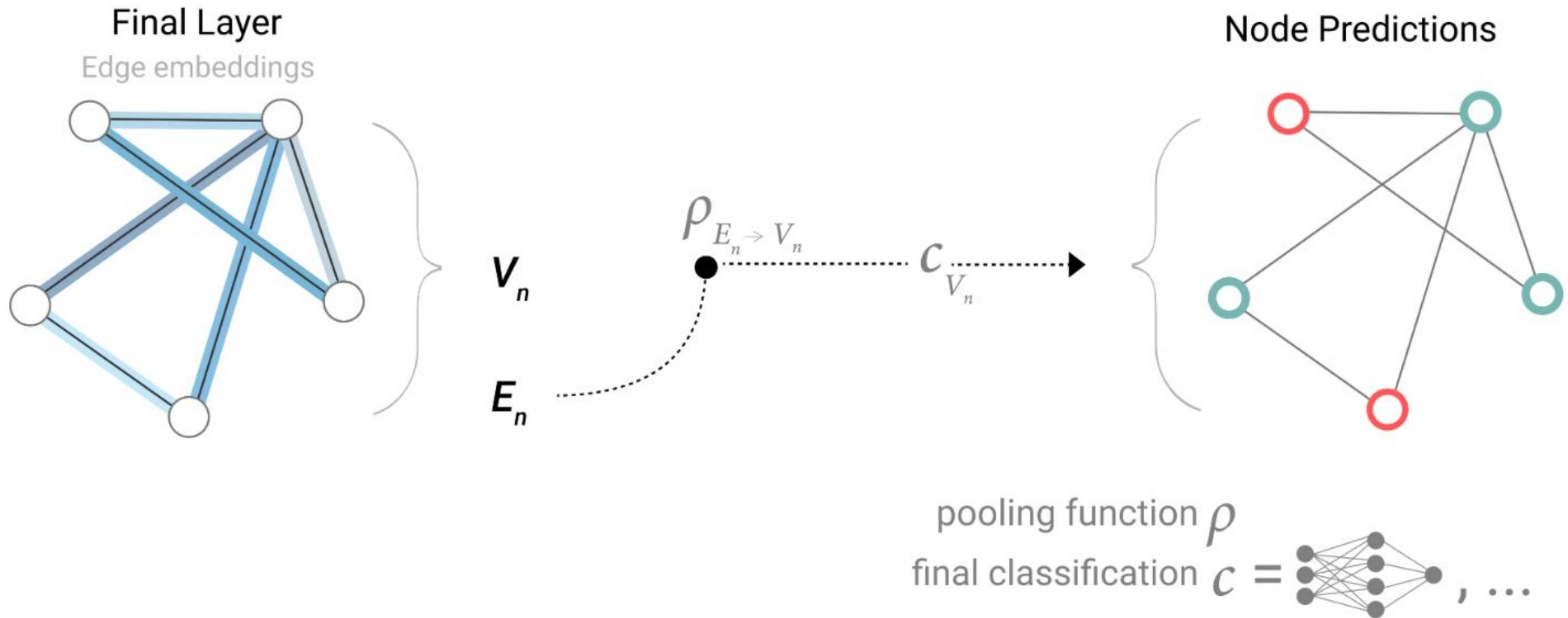
We can learn neighbor interactions through pooling!



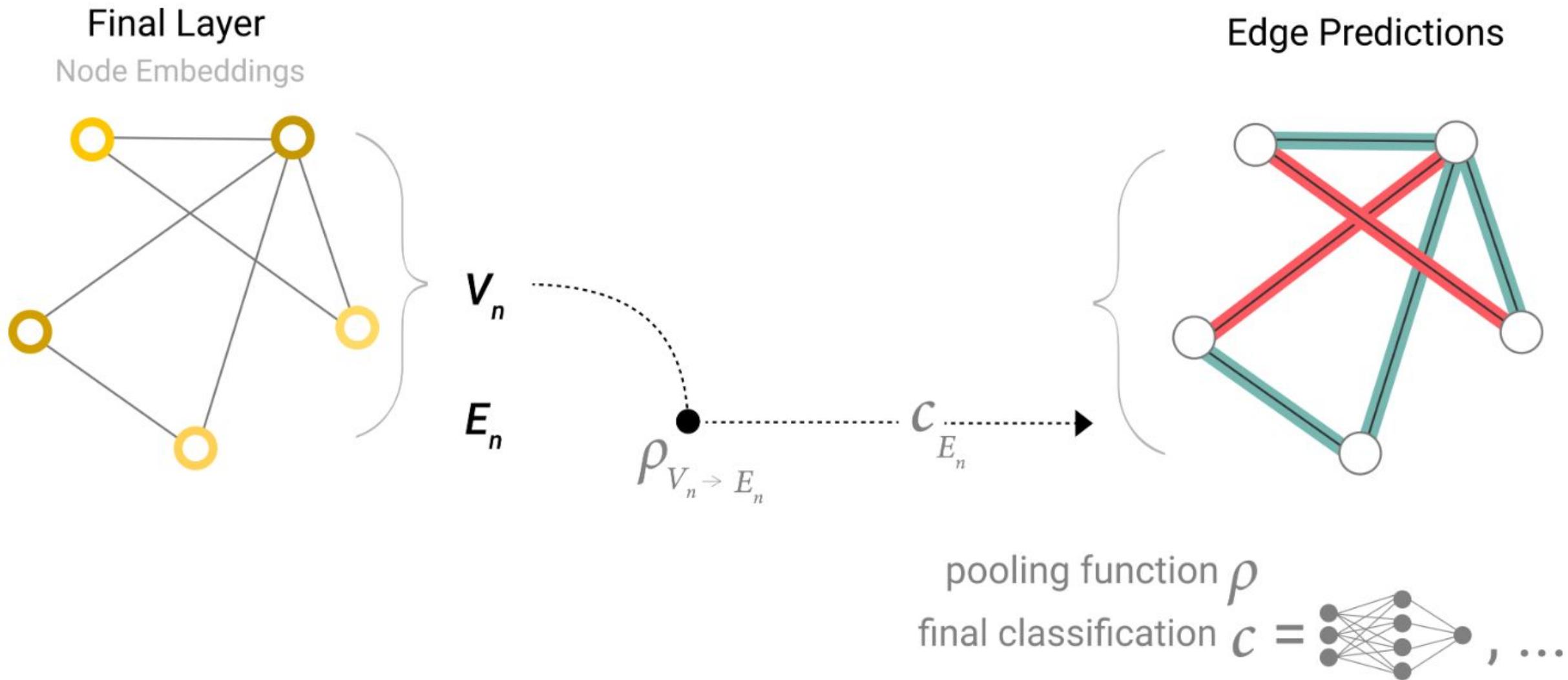
A matrix of edge vectors is created and these are aggregated via sum operation to update node value

$$\rho_{E_n \rightarrow V_n}$$

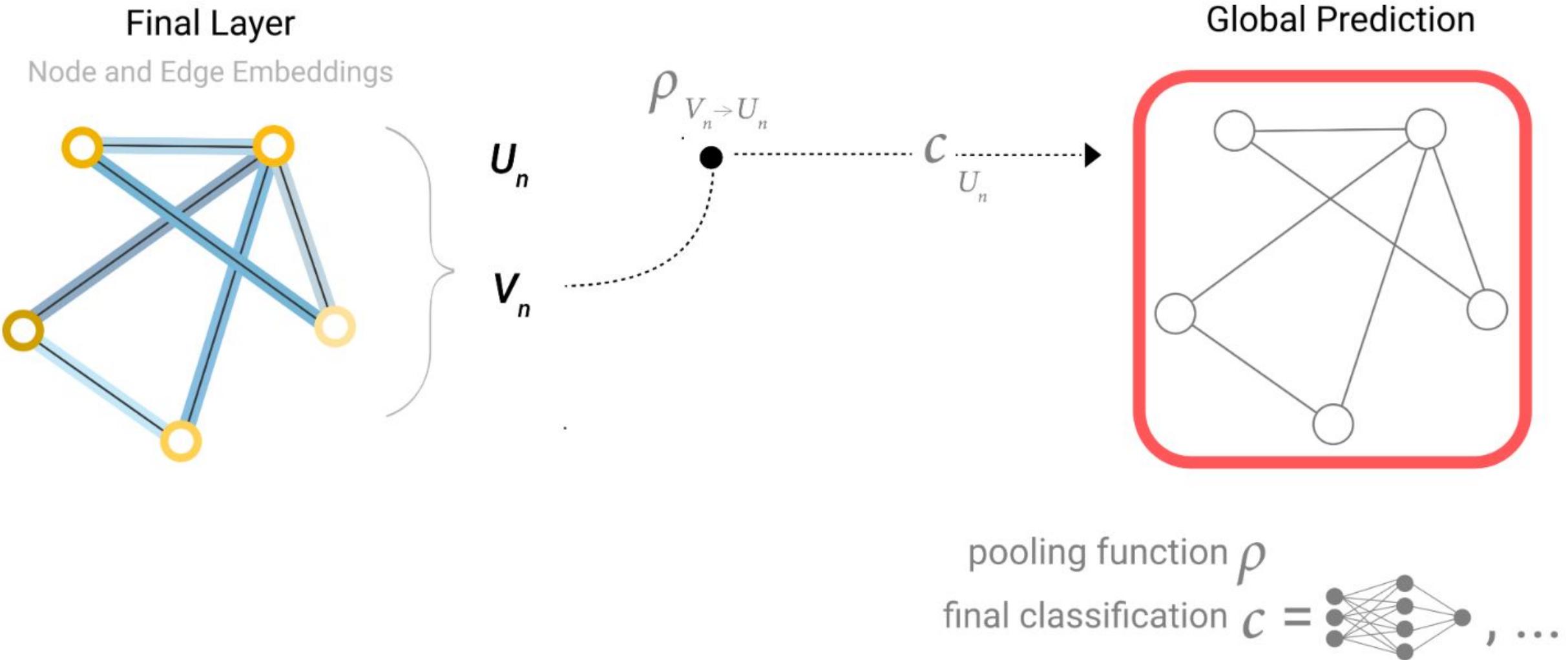
Using pooling we can predict, node, edge, or global attributes



Using pooling we can predict, node, edge, or global attributes

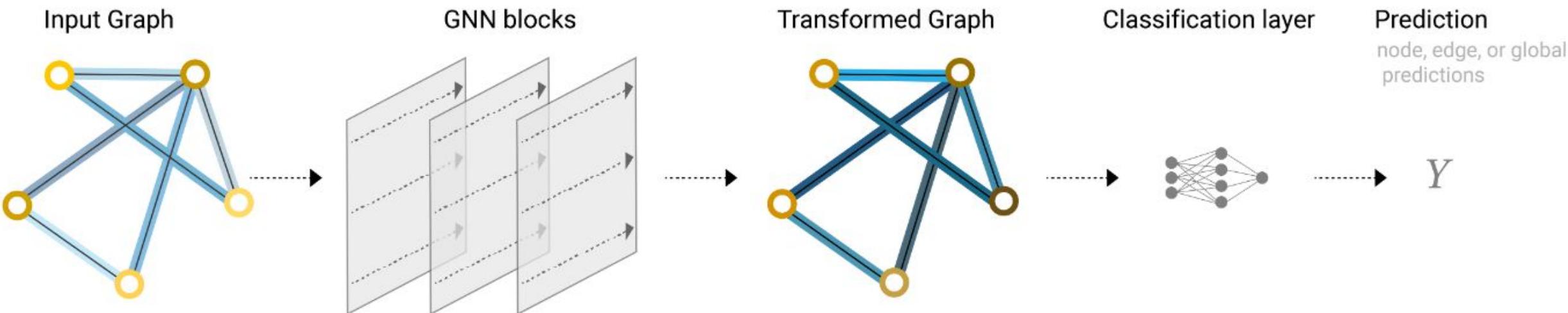


Using pooling we can predict, node, edge, or global attributes



Simple GNNs do not use connectivity in calculations

Each node , edge, and graph attribute is processed independently and connectivity is only used during pooling



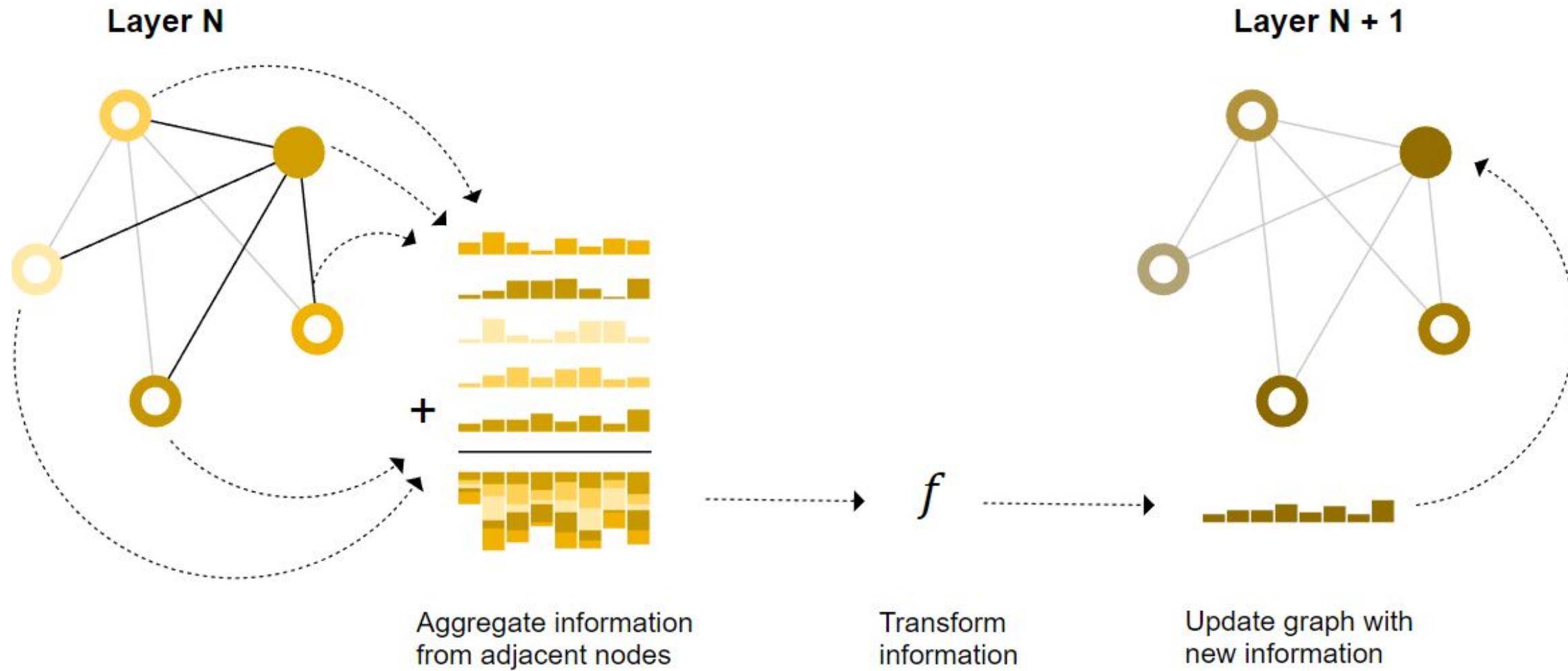
An end-to-end prediction task with a GNN model.

Message passing allows connectivity to influence embeddings

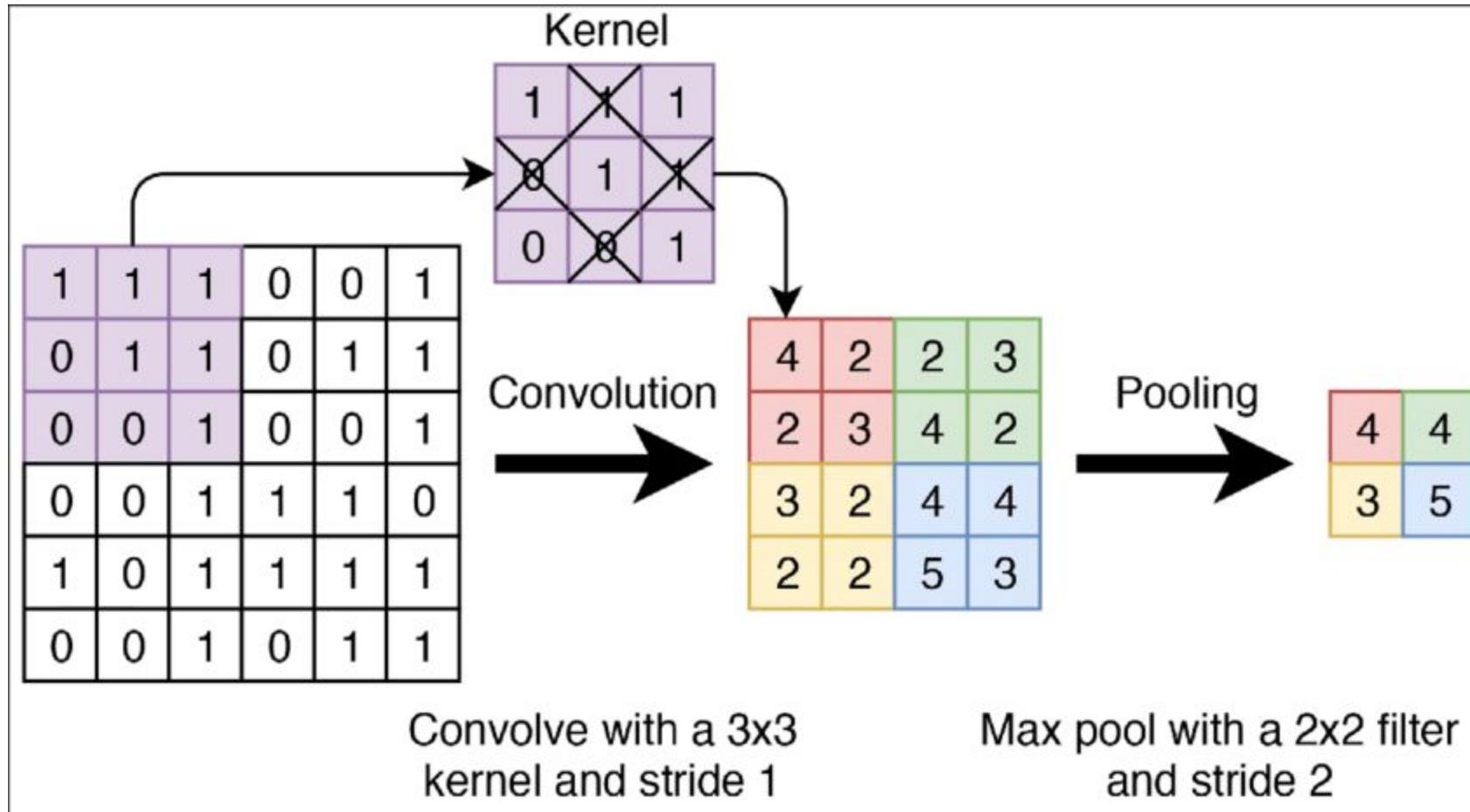
1. For each node in the graph, gather all the neighboring node embeddings (or messages)
2. Aggregate all messages via an aggregate function (like sum)
3. All pooled messages are passed through an update function, usually a learned neural network

Gilmer et al “Neural Message Passing for Quantum Chemistry” Proc 34th International Conference on Machine Learning, 2017.

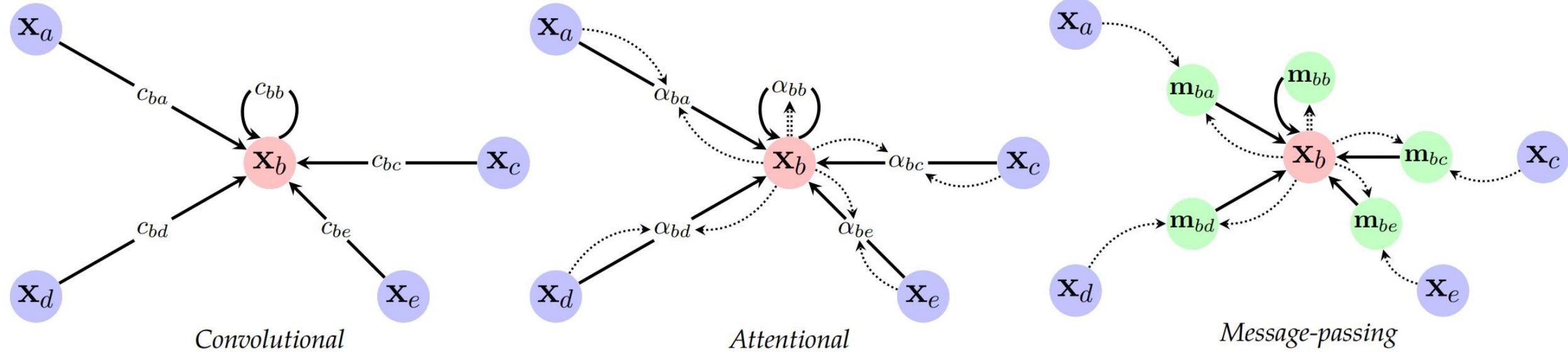
Message passing allows connectivity to influence embeddings



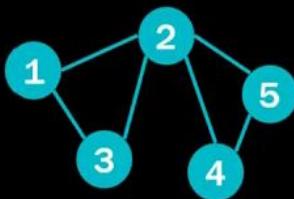
Message passing is similar to convolutional neural networks



We can use connectivity in different ways



Message passing learns from all connected nodes



$$h'_i = \sigma\left(\sum_{j \in N(i)} W * h_j\right)$$

1	1	1	0	0
1	1	1	1	1
1	1	1	0	0
0	1	0	1	1
0	1	0	1	1

h_1

h_2

h_3

h_4

h_5

adjacency matrix

features per node

[5, 5]

[5, 4]

[4, 8]

[5, 8]

learnable weight matrix

h'_1

h'_2

h'_3

h'_4

h'_5

embedding per node

Velickovik, “Graph attention networks” ICLR 2018

DeepFindR “Understanding graph attention networks”, YouTube 2021

We know that some connections are more important than others



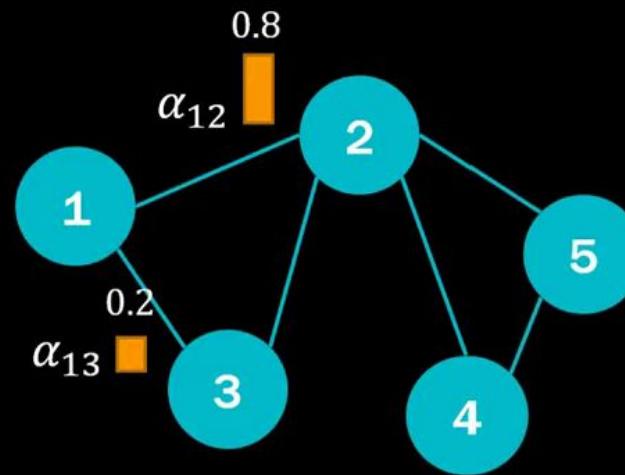
Velickovic, “Graph attention networks” ICLR 2018

DeepFindR “Understanding graph attention networks”, YouTube 2021

We can learn which nodes to pay attention to over others

$$e_{ij} = \text{a}(Wh_i, Wh_j)$$

attention coefficient



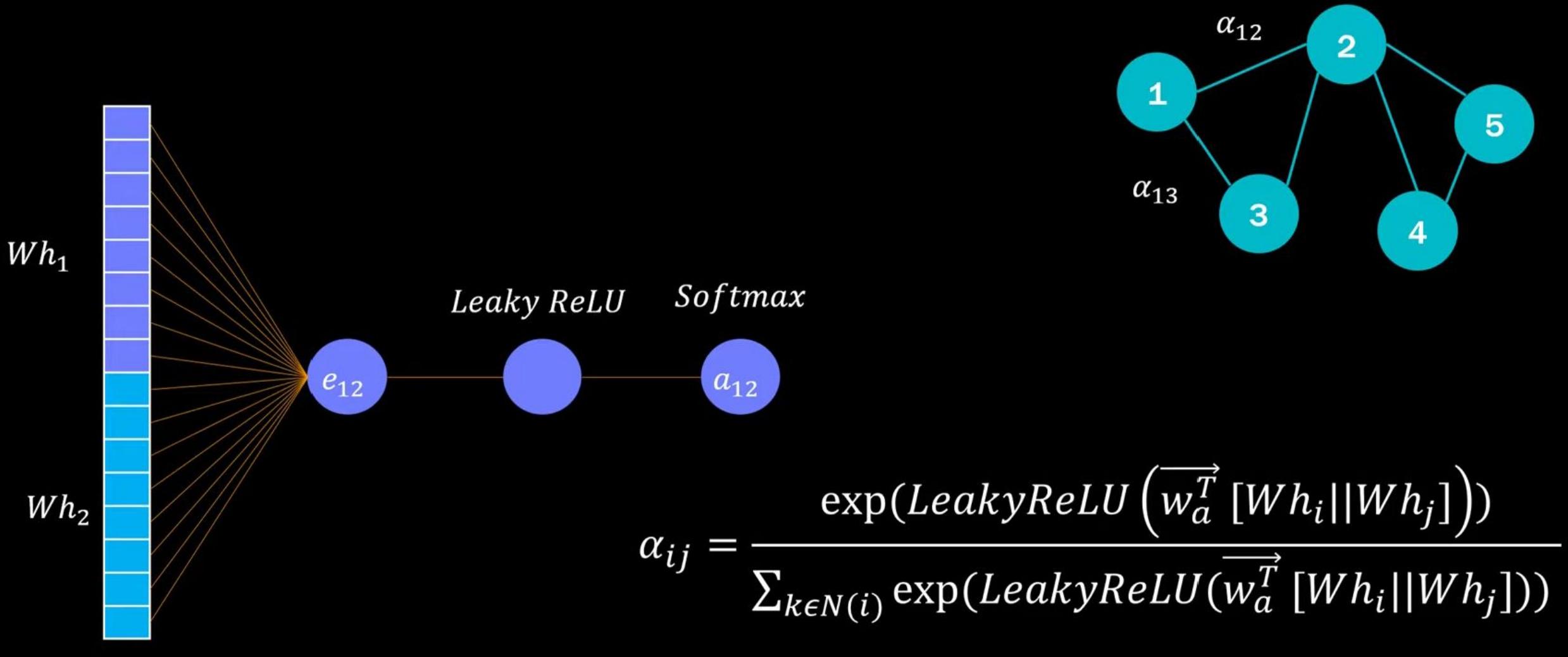
$$\alpha_{ij} = \text{softmax}_j(e_{ij}) = \frac{\exp(e_{ij})}{\sum_{k \in N(i)} \exp(e_{ik})} = \frac{\exp(a(Wh_i, Wh_j))}{\sum_{k \in N(i)} \exp(a(Wh_i, Wh_j))}$$

normalized attention coefficient

Velickovic, "Graph attention networks" ICLR 2018

DeepFindR "Understanding graph attention networks", YouTube 2021

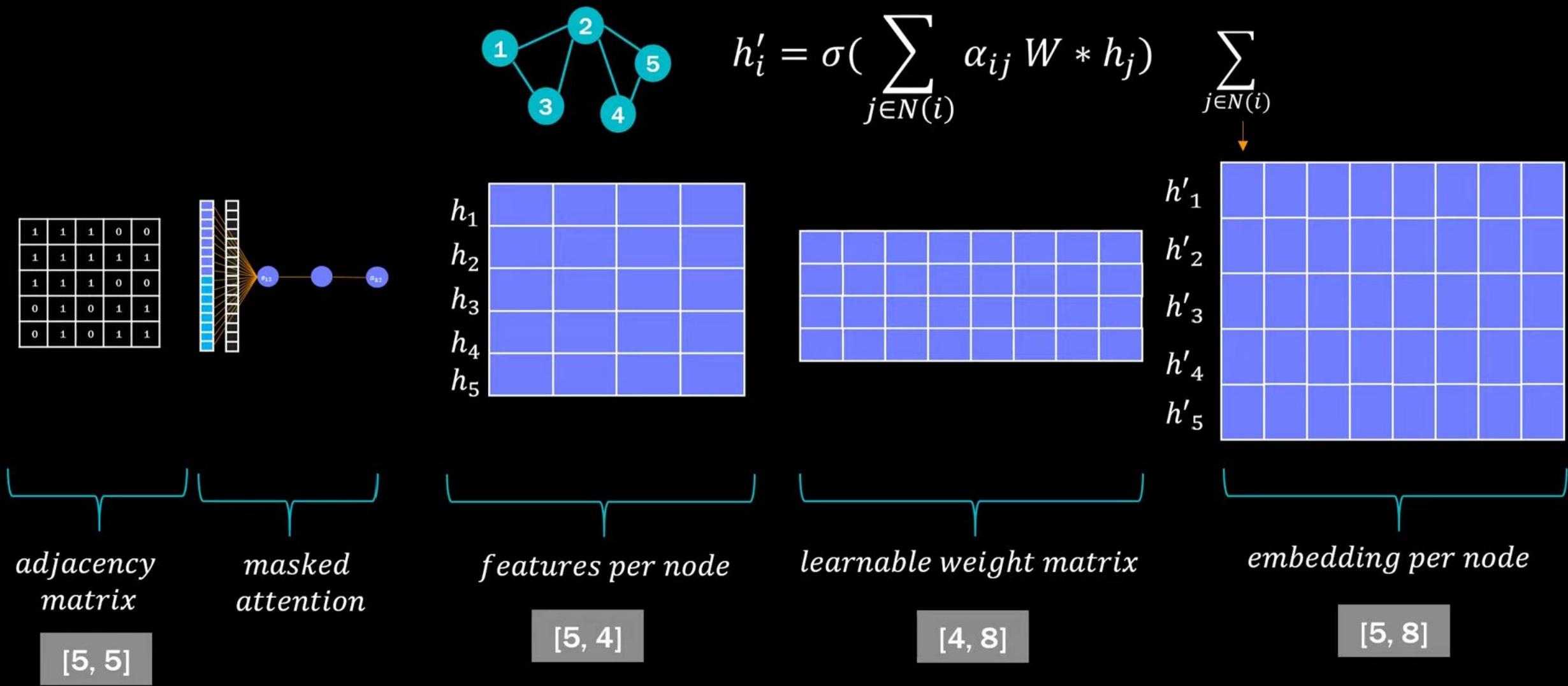
Attention coefficients are learned for all connections



Velickovic, "Graph attention networks" ICLR 2018

DeepFindR "Understanding graph attention networks", YouTube 2021

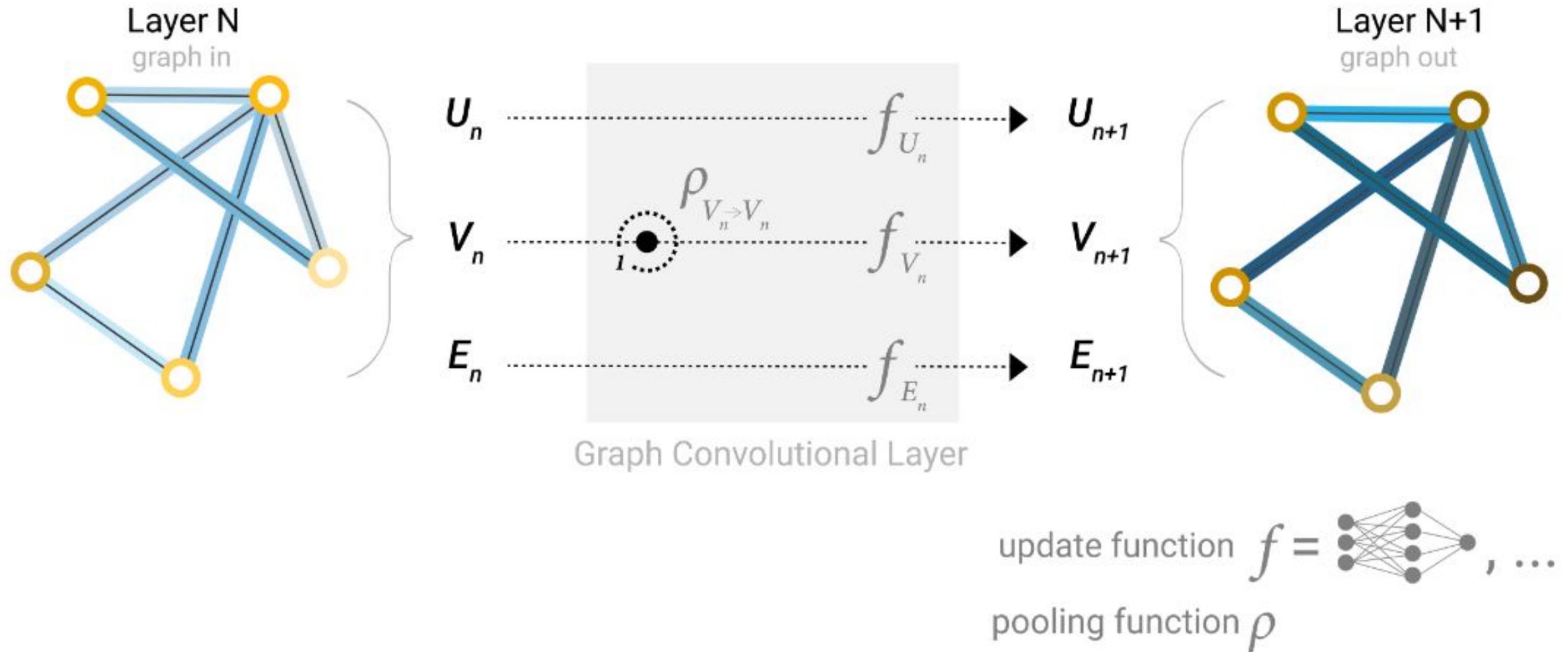
Overall, it looks similar to message passing, but with more learning



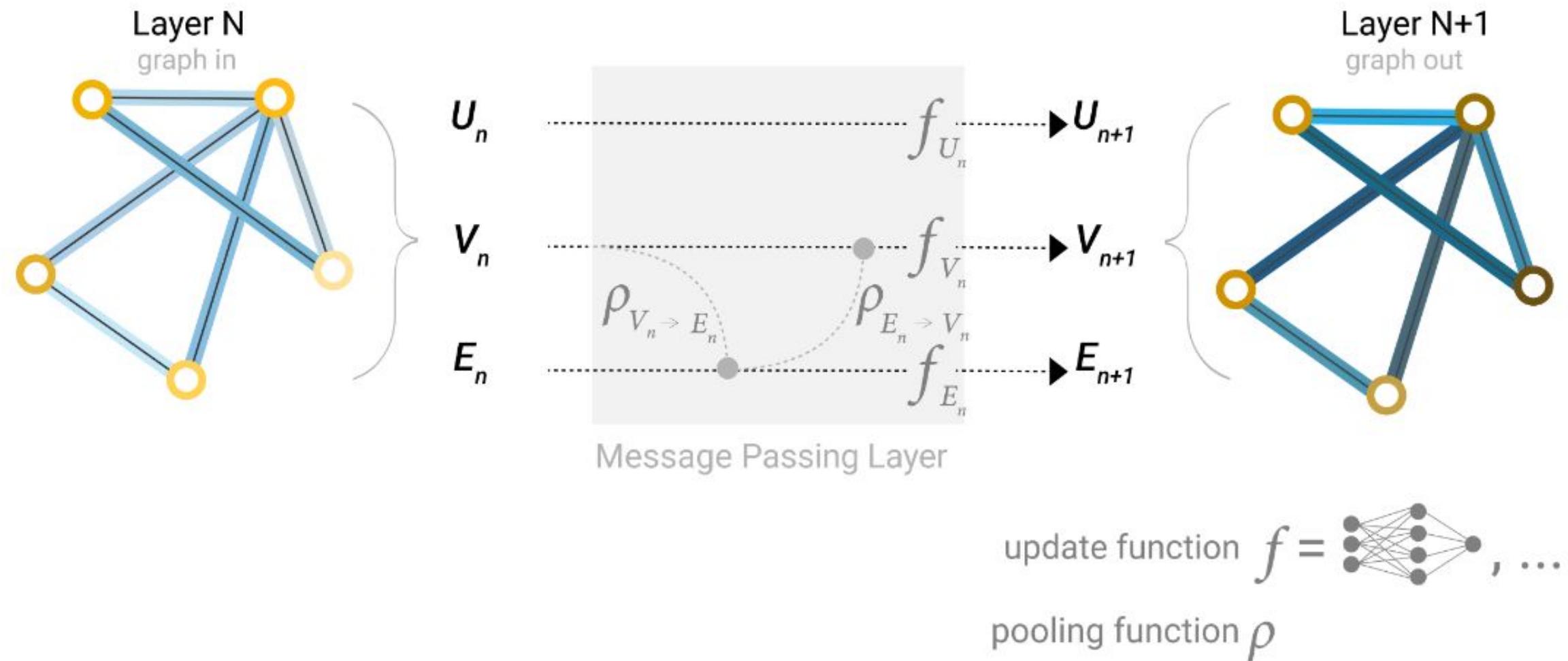
Velickovik, "Graph attention networks" ICLR 2018

DeepFindR "Understanding graph attention networks", YouTube 2021

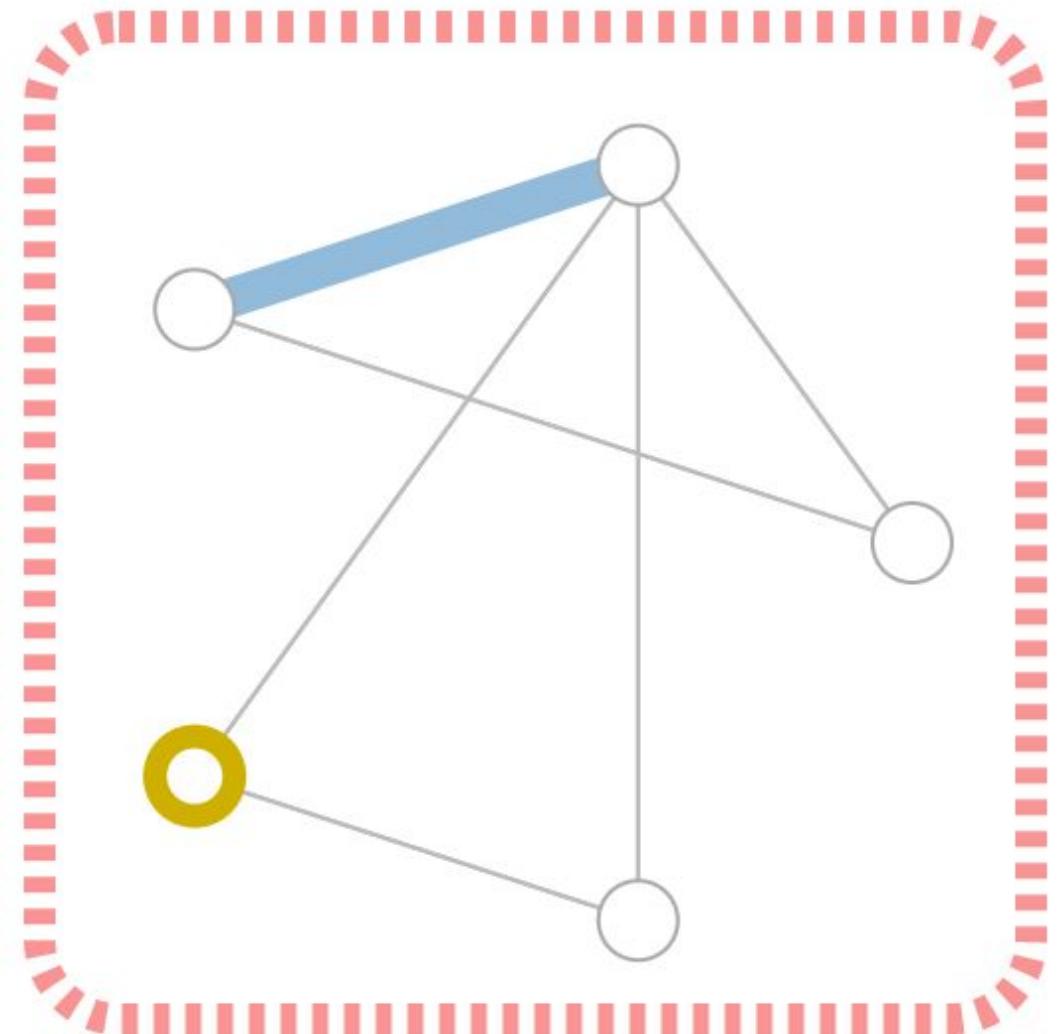
Repeated layers expands information from across graph



We can also go between components with message passing



What to we do if our edge/node embeddings are different shape?



Vertex (or node) embedding



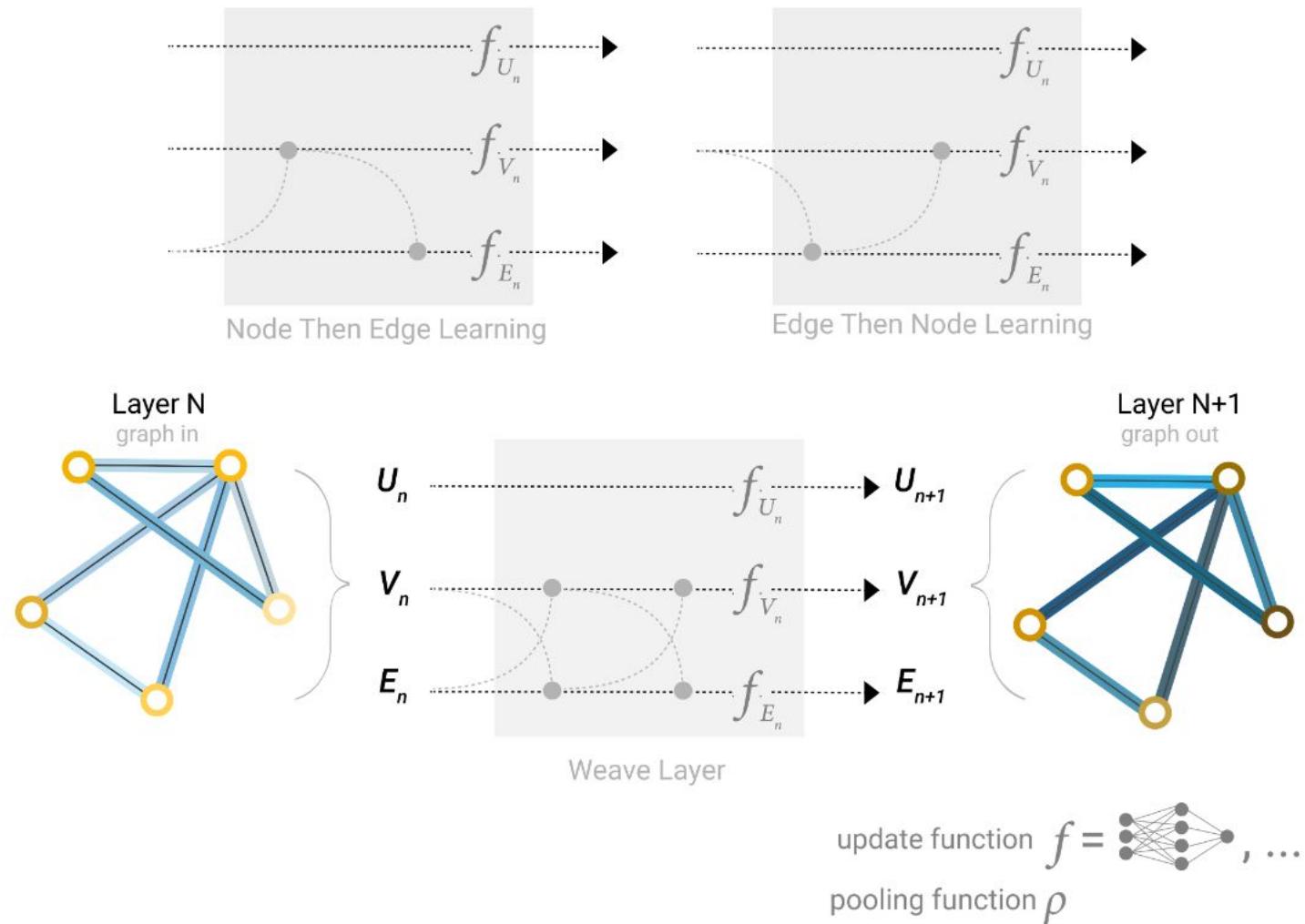
Edge (or link) attributes and embedding



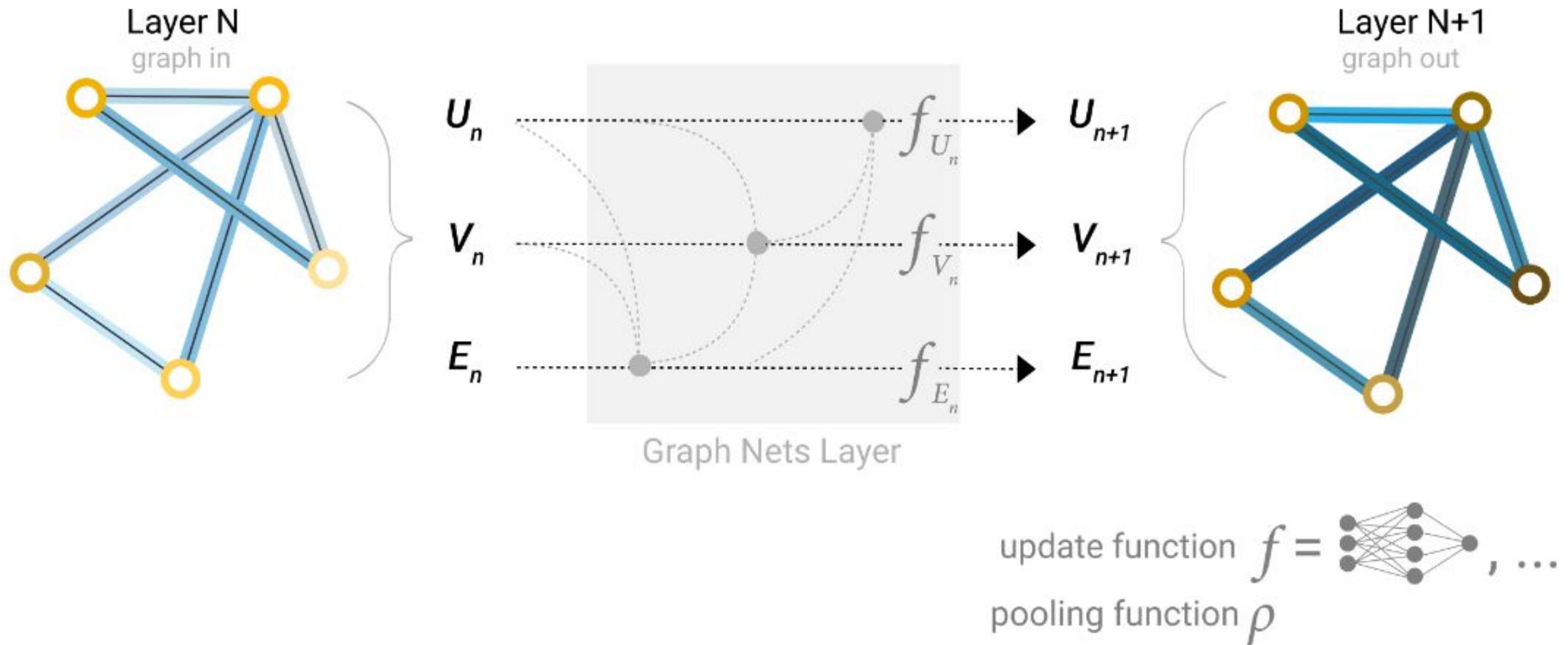
Global (or master node) embedding



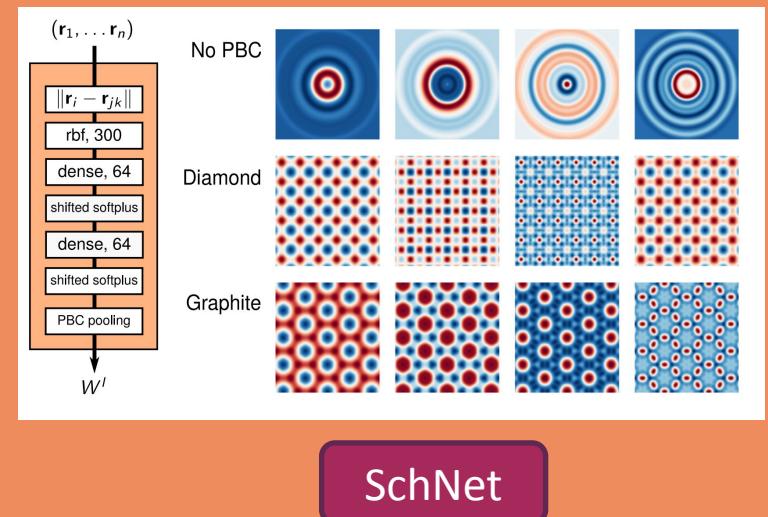
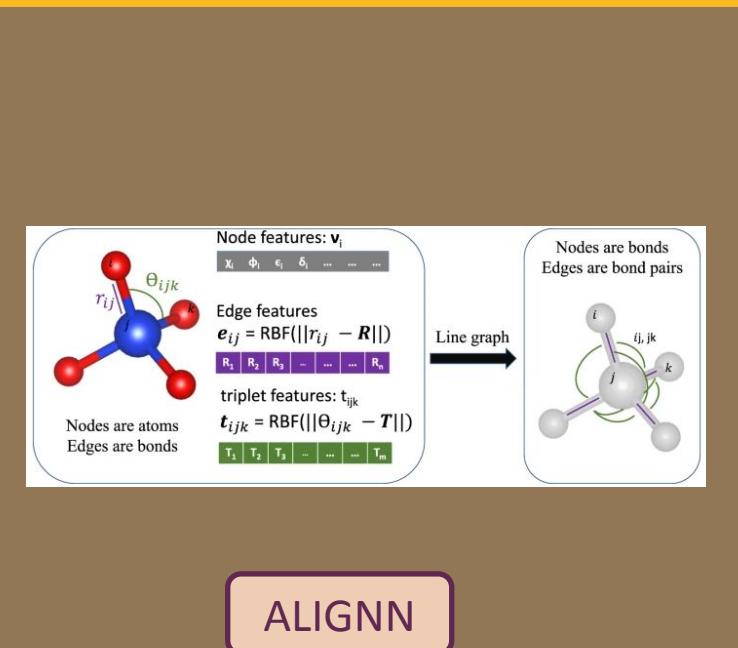
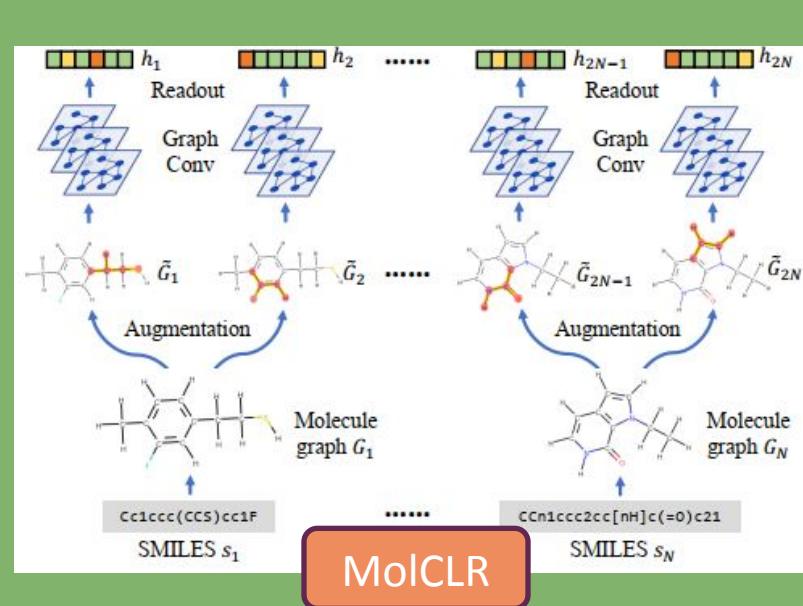
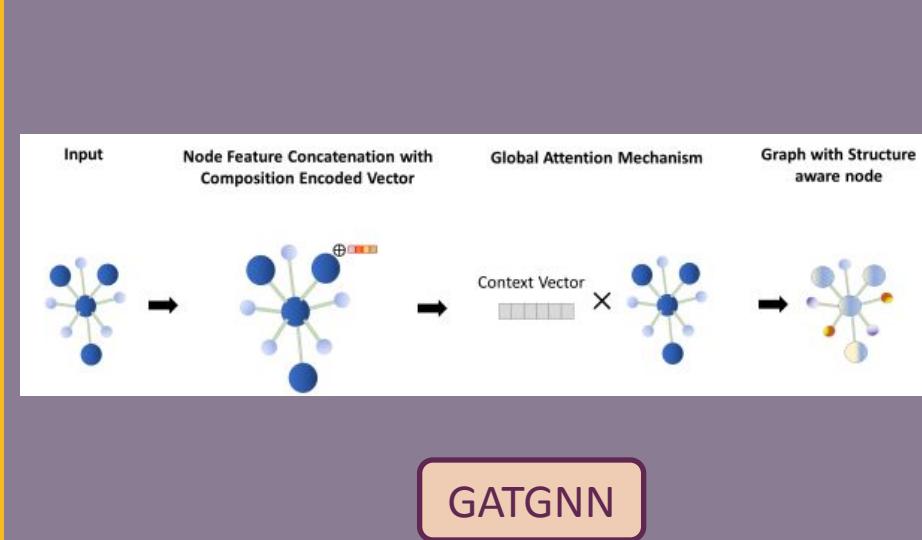
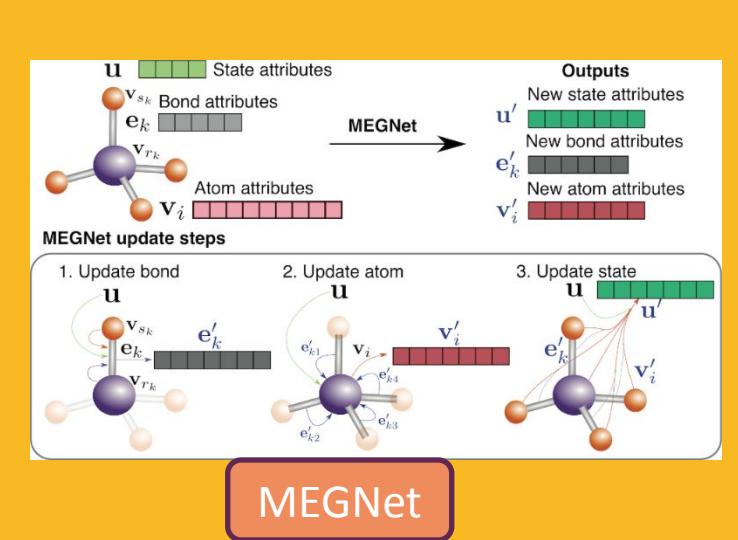
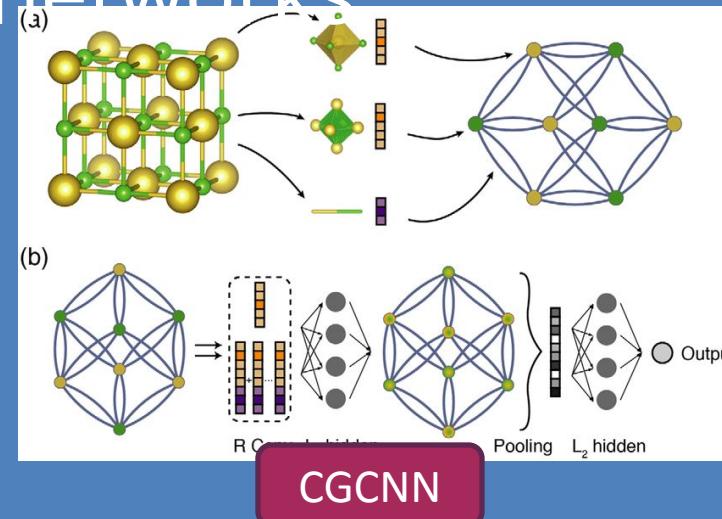
We have many options and choices when it comes to order



Global attribute acts as a “master node” to reduce graph distances



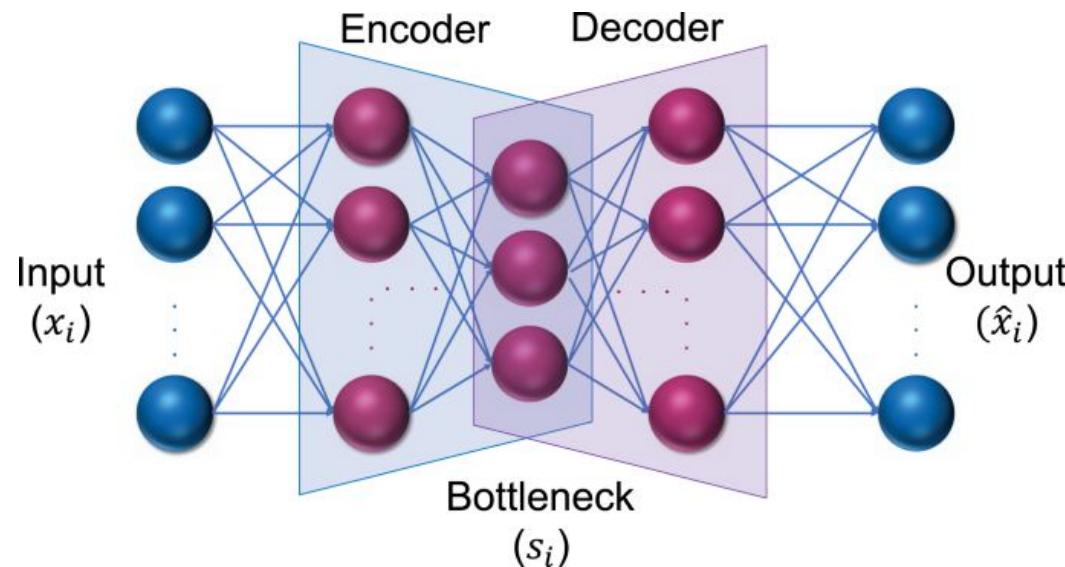
Some of the best materials informatics tools are graph networks



What do new model architectures give the community?

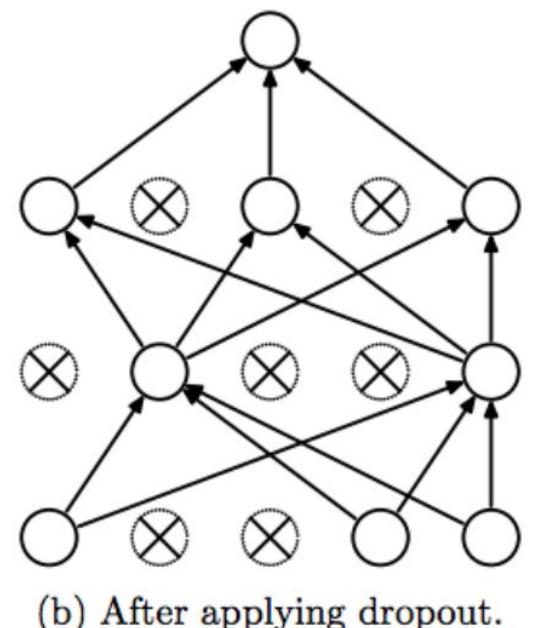
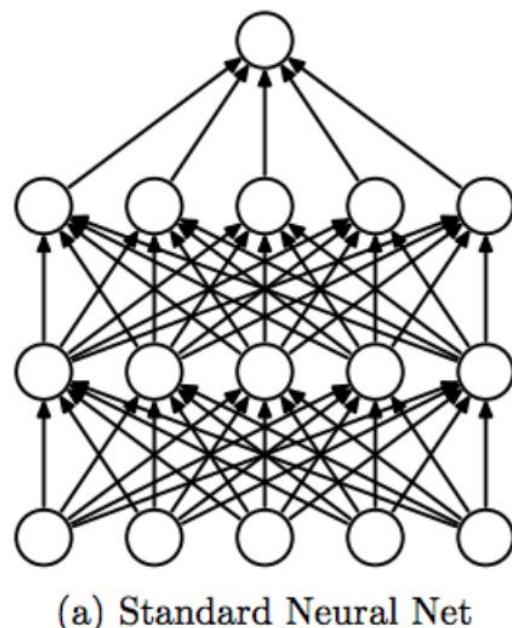


Is feature reduction sufficient? Or is something else needed?



Dilution (dropout)
regularization

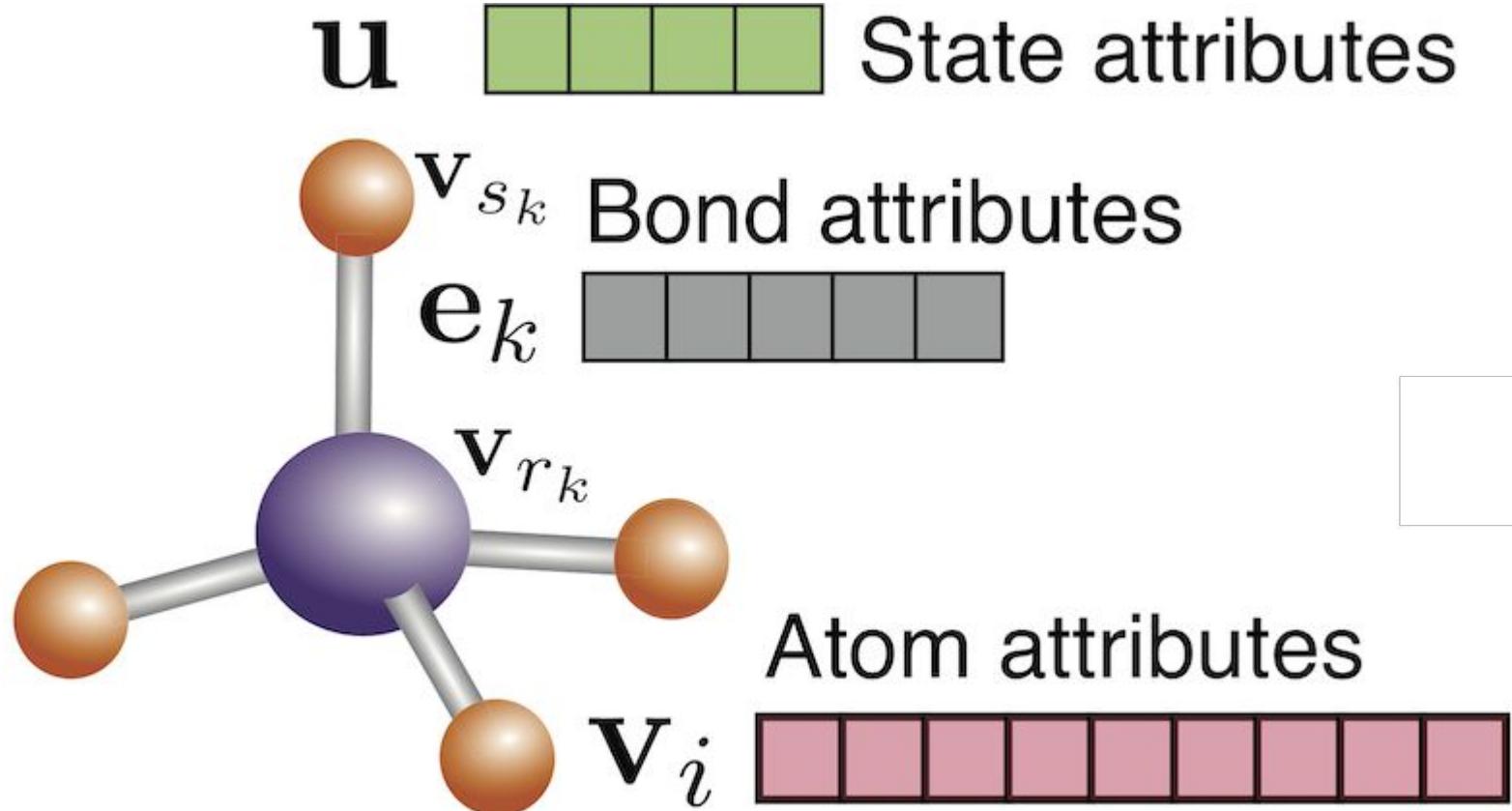
Feature reduction



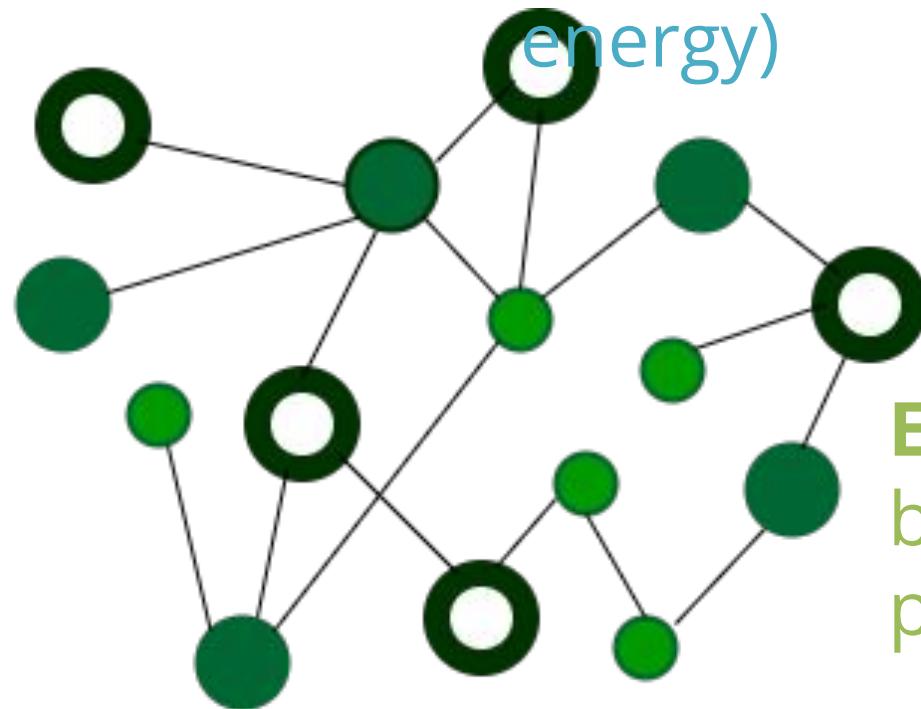
Features are becoming quite large for many GNNs



Features are becoming quite large for many GNNs



Is there a better, simpler graph representation?



Nodes: atoms with limited features
(quantum numbers, ground state
energy)

Edges: bonds, limited connectivity,
based on physically meaningful
parameters

Graph: centrality, clustering coefficient,
connectivity, motifs, symmetry

2 pt statistics

