Graphs are everywhere

1 **Graphs are everywhere**

- Complex data structures
- Basics of graph theory

2 **Learning on Graphs**

- Graph embedding
- Transductive and inductive learning
- Tasks on graph learning

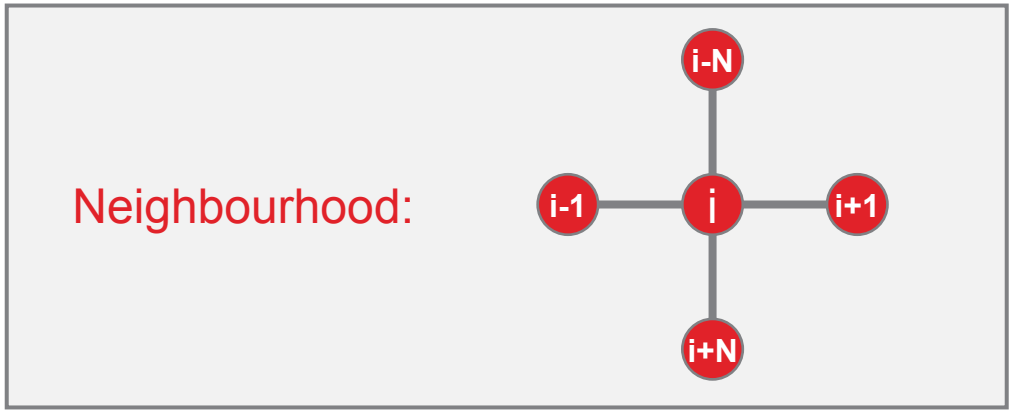
3 **A few examples**

- Taxonomy of methods
- Graph convolution
- Message passing
- Graph Transformer

Data structures: Ordered structures

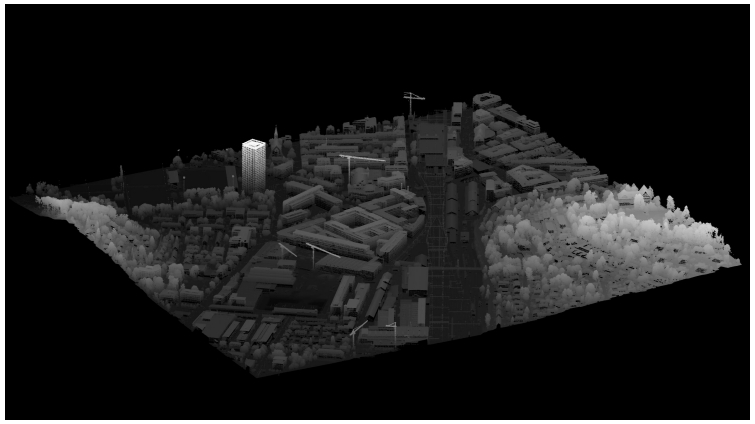
Highly ordered data

Rebirth of Deep learning was thanks to pictures, text and speech recognition

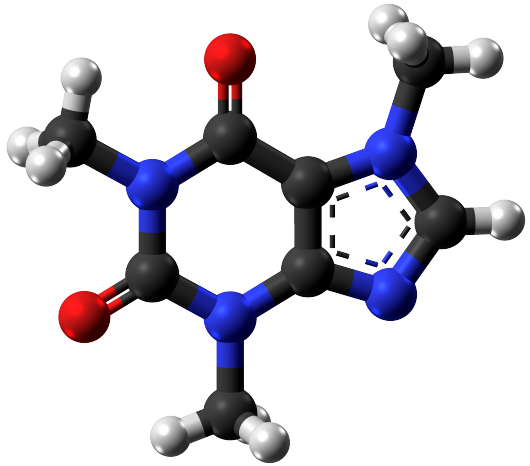


Data structures: Disordered structures

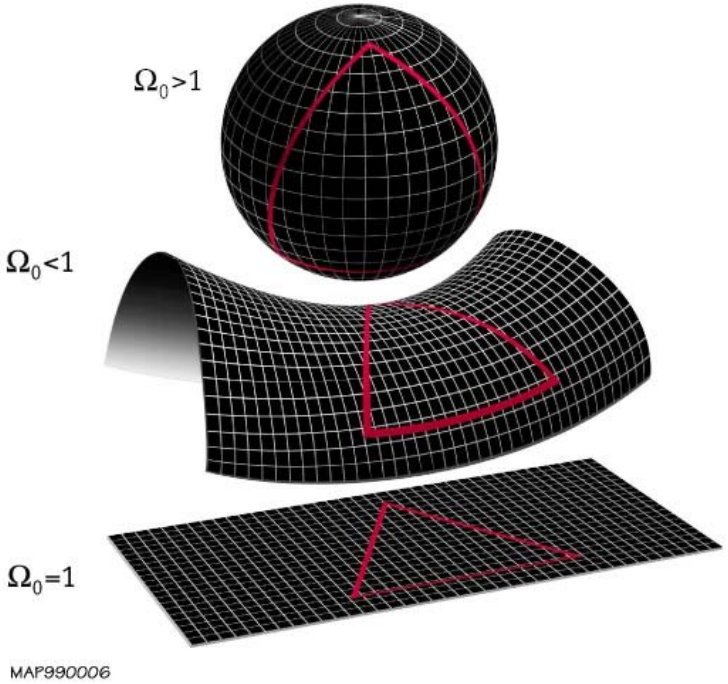
LIDAR

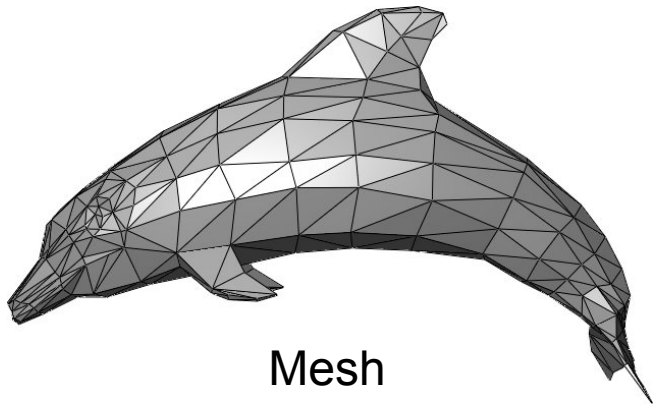


Molecule



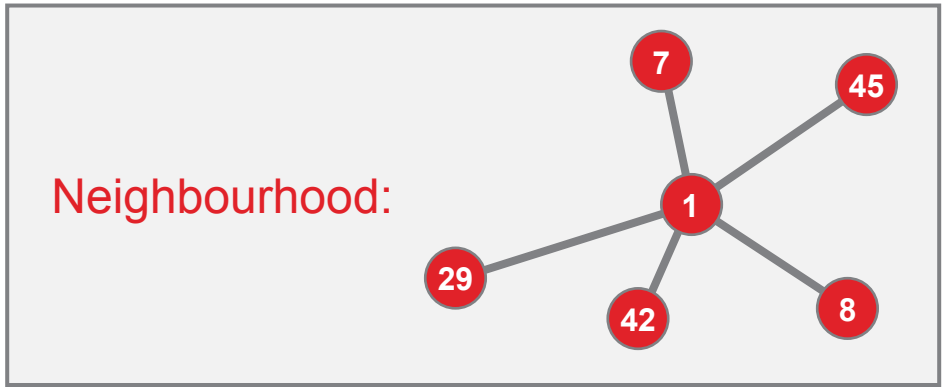
Complex geometries





Mesh

Geometric Deep Learning



Michael M. Bronstein, Joan Bruna, Taco Cohen, Petar Veličković,
Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges https://arxiv.org/abs/2104.13478

Data structures: Non spatial structures

Social network

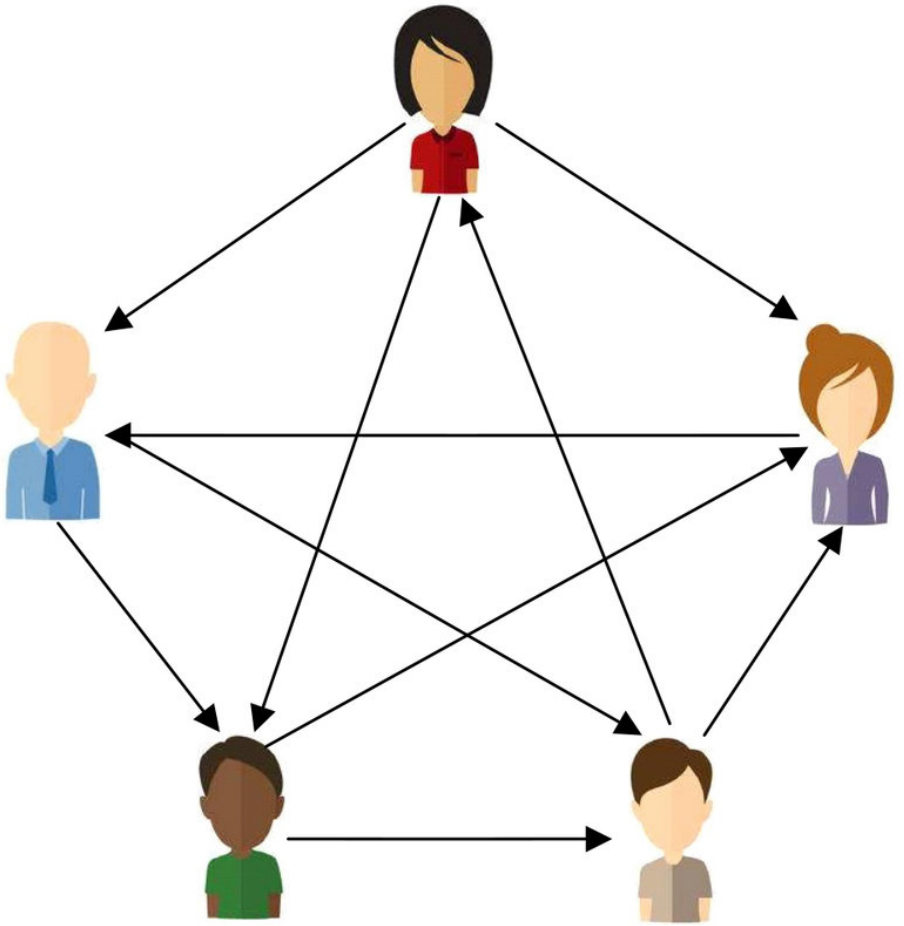
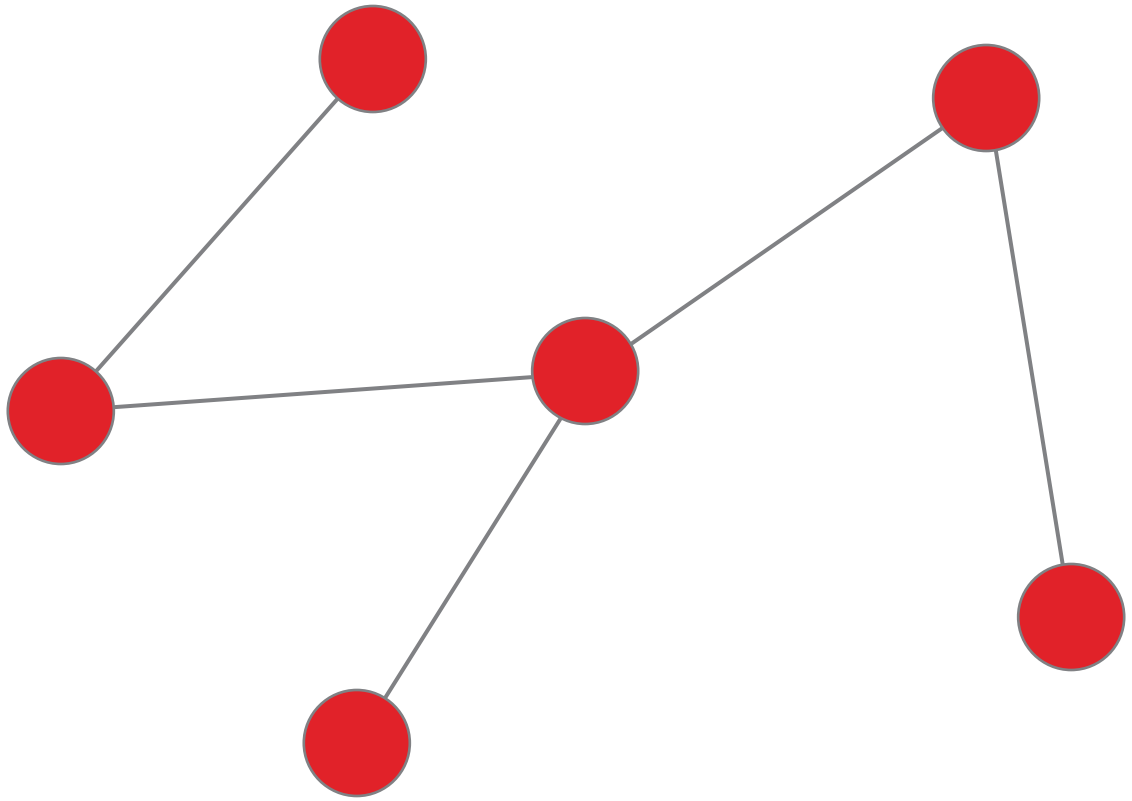


Illustration from Liu, Yujia & Liang, Changyong & Chiclana, Francisco & Wu, Jian. (2020). A knowledge coverage-based trust propagation for recommendation mechanism in social network group decision making. Applied Soft Computing. 101. 10.1016/j.asoc.2020.107005.

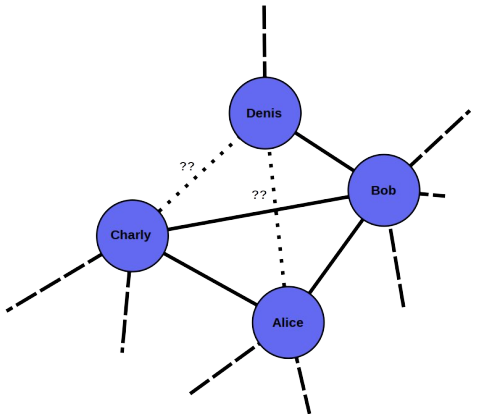
First definition



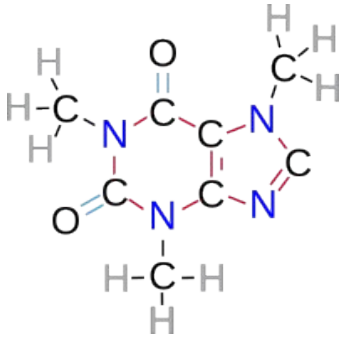
A graph is a set of interconnected entities

Graphs are everywhere

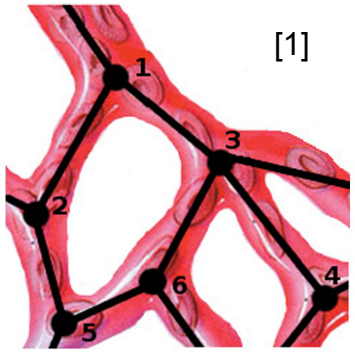
Social networks



Molecules

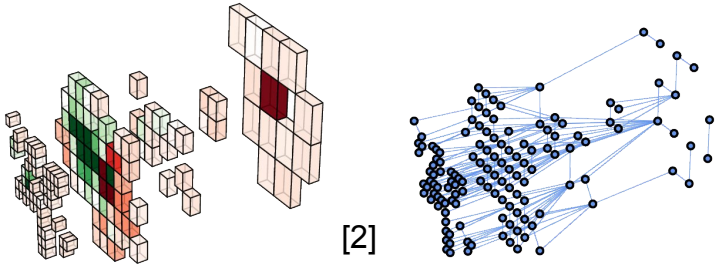


Capillary networks



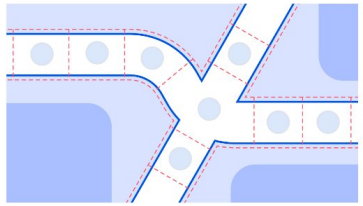
[1]

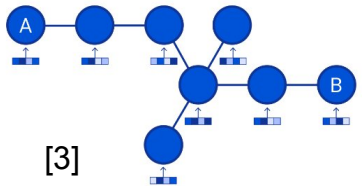
Particle physics



[2]

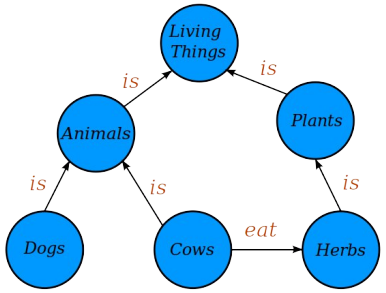
Itinerary recommendations





[3]

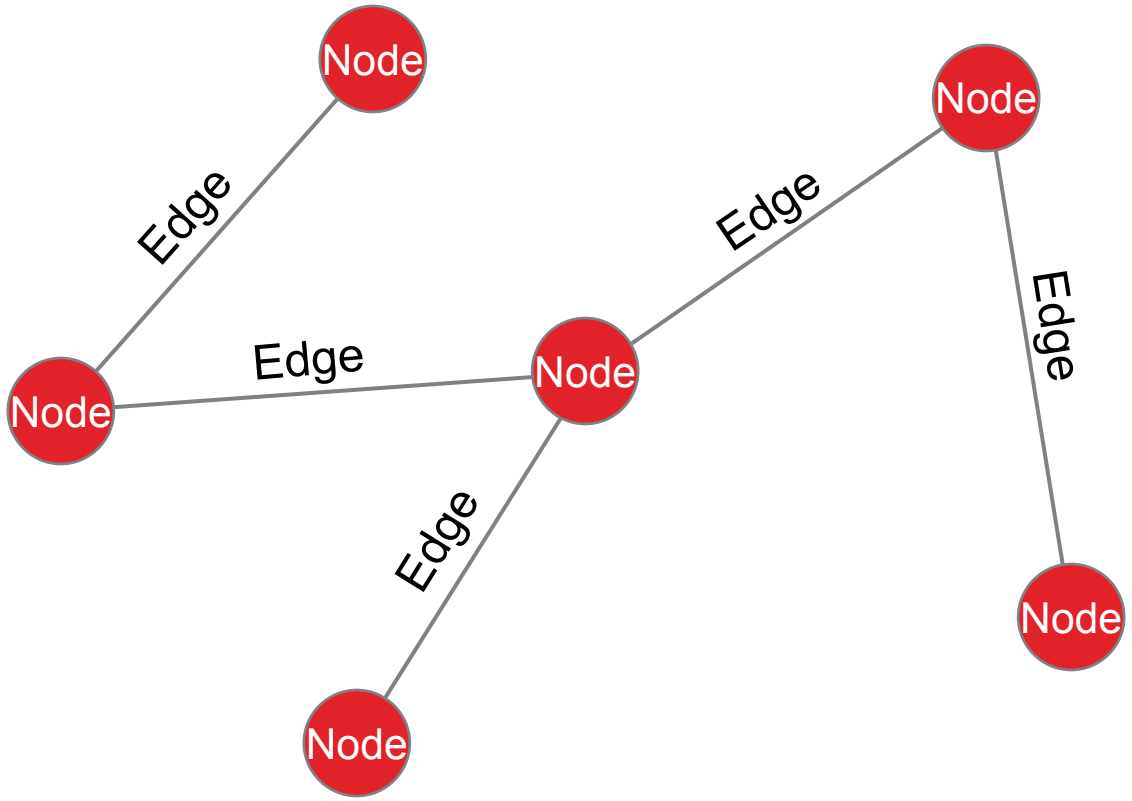
Knowledge graphs



And many other fields: biology, recommendation systems, computer vision, robotics, medical diagnosis,...

[1] Erbertseder et al. (2012). A coupled discrete/continuum model for describing cancer-therapeutic transport in the lung. PLoS One, 7(3), e31966.
[2] Shlomi et al., “Graph neural networks in particle physics,” Mach. Learn.: Sci. Technol., vol. 2, no. 2, p. 021001, Jan. 2021, doi: 10.1088/2632-2153/abbf9a.
[3] Derrow-Pinion et al., “ETA Prediction with Graph Neural Networks in Google Maps,” in Proceedings of the 30th ACM International Conference on Information & Knowledge Management New York, NY, USA, Oct. 2021, pp. 3767–3776. doi: 10.1145/3459637.3481916.

Vocabulary

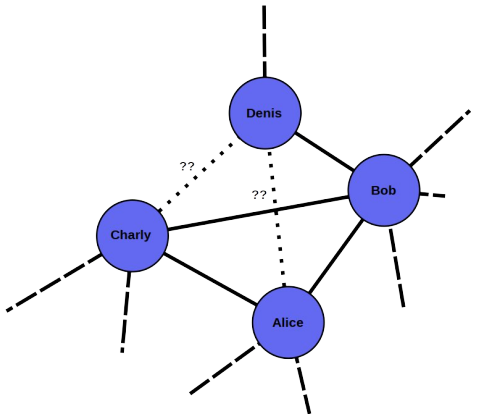


Node = vertex

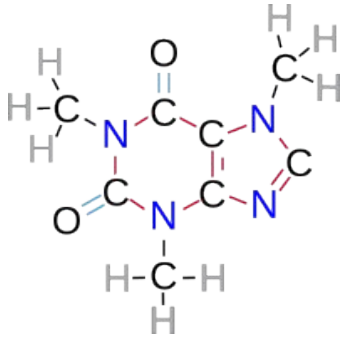
Graph

Vocabulary: Node / vertex

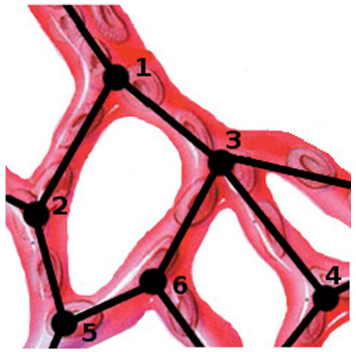
Persons



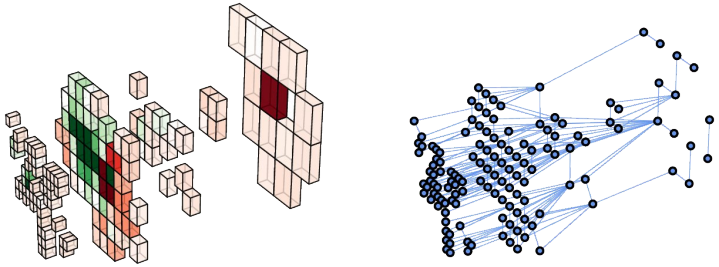
Atoms



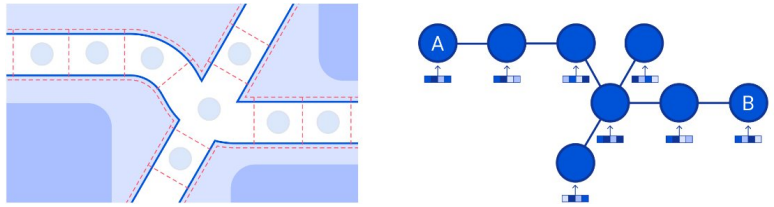
Intersections



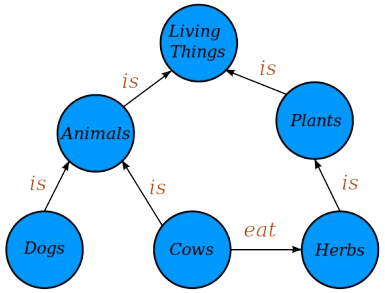
Particles



Road sections



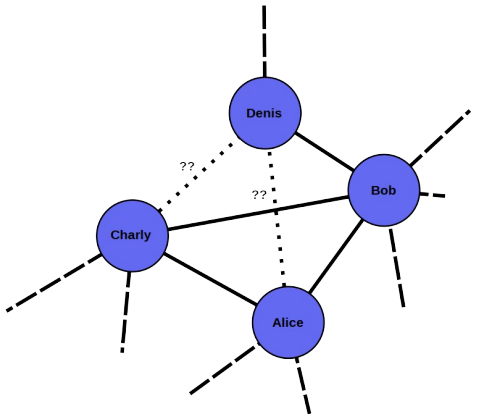
Concepts



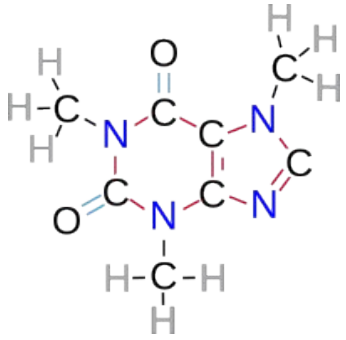
In many other fields: an aminoacid in a protein (biology), a customer (recommendation systems), an object in a picture (computer vision), brain regions on MRI (medical diagnosis), joints (robotics),...

Vocabulary: Edge

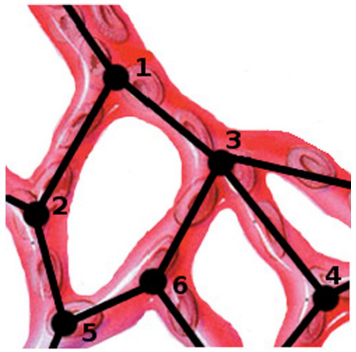
Relationship



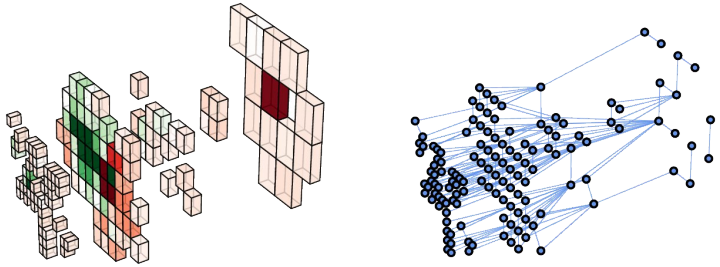
Type of bond



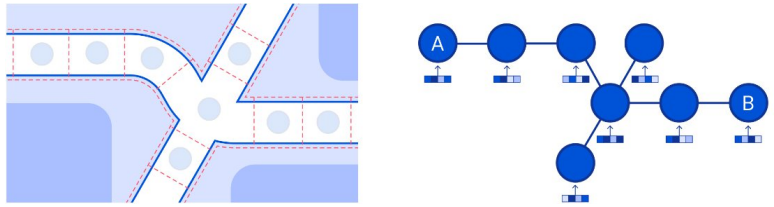
Vessel



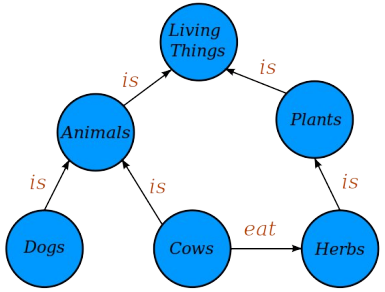
Decayed of



Time



Statement

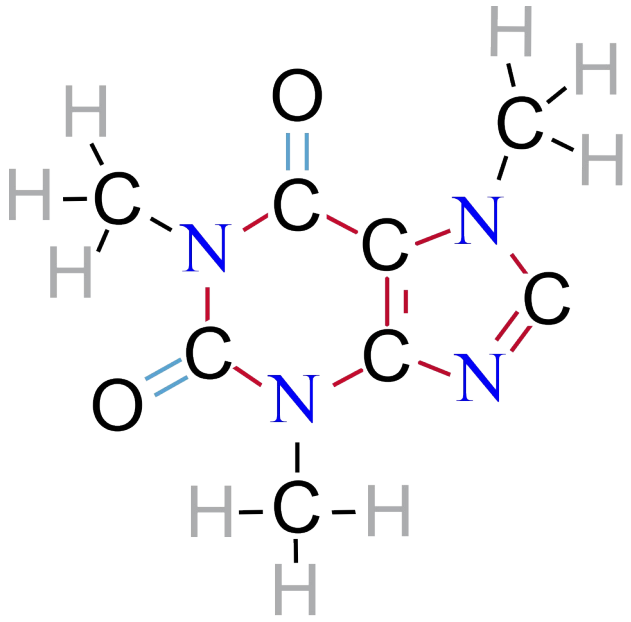


In many other fields: distance between residues (biology), connected customers (recommendation systems), interaction between objects (computer vision), interaction between brain regions (medical diagnosis), connection between joints (robotics),...

Vocabulary: Directed or undirected graph

A relationship (edge) can be symmetrical or not between nodes

Undirected graphs

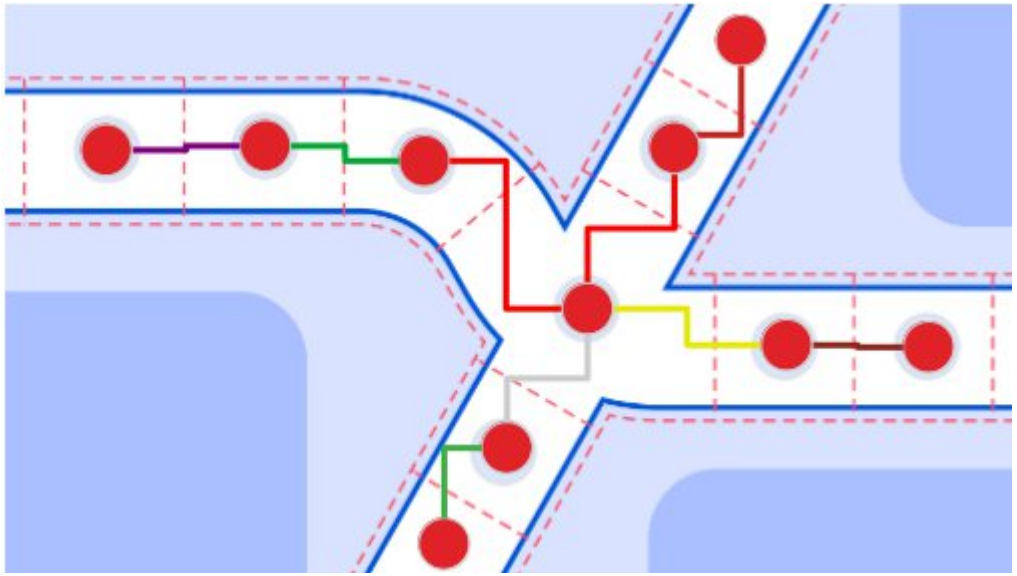


Directed graphs

*Alterna2 http://www.alterna2.com, CC BY 2.0 <https://creativecommons.org/licenses/by/2.0>, via Wikimedia Commons

Vocabulary: Weight

Edges can carry information → edge **weight**



Time

Graphs store information: Features

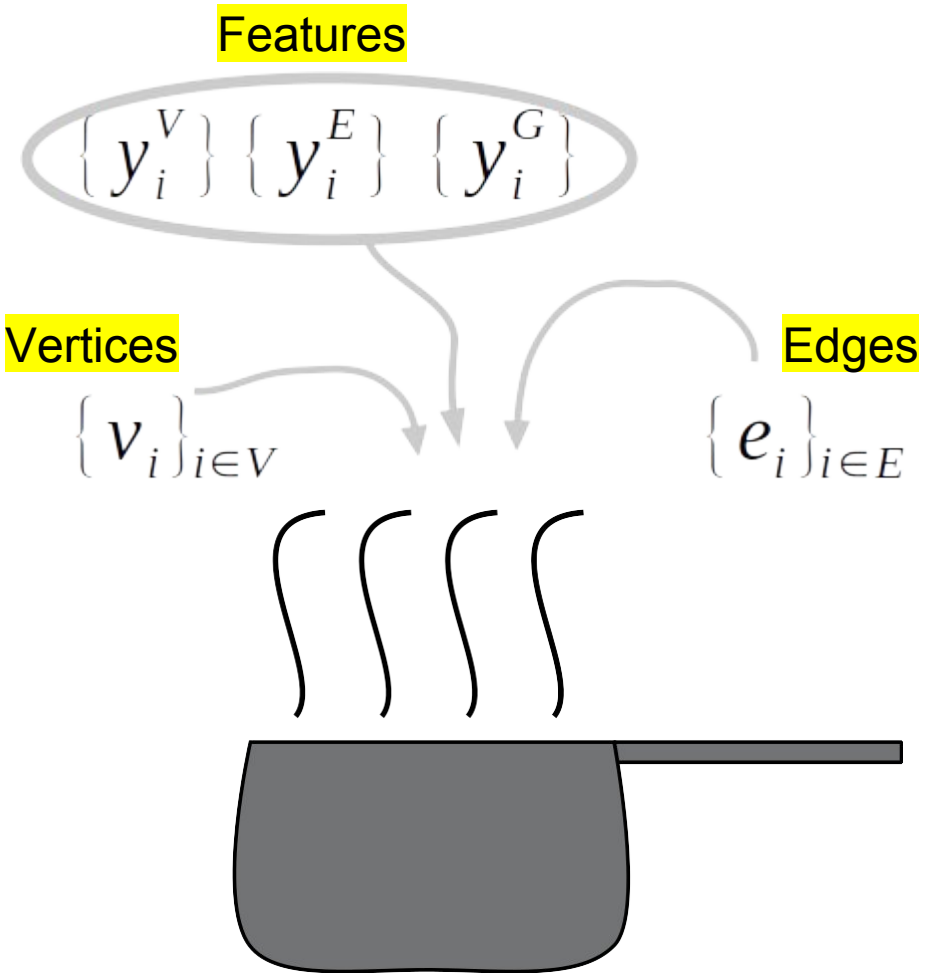
Graphs can store information on **node**, **edge** or **globally (on graph)**

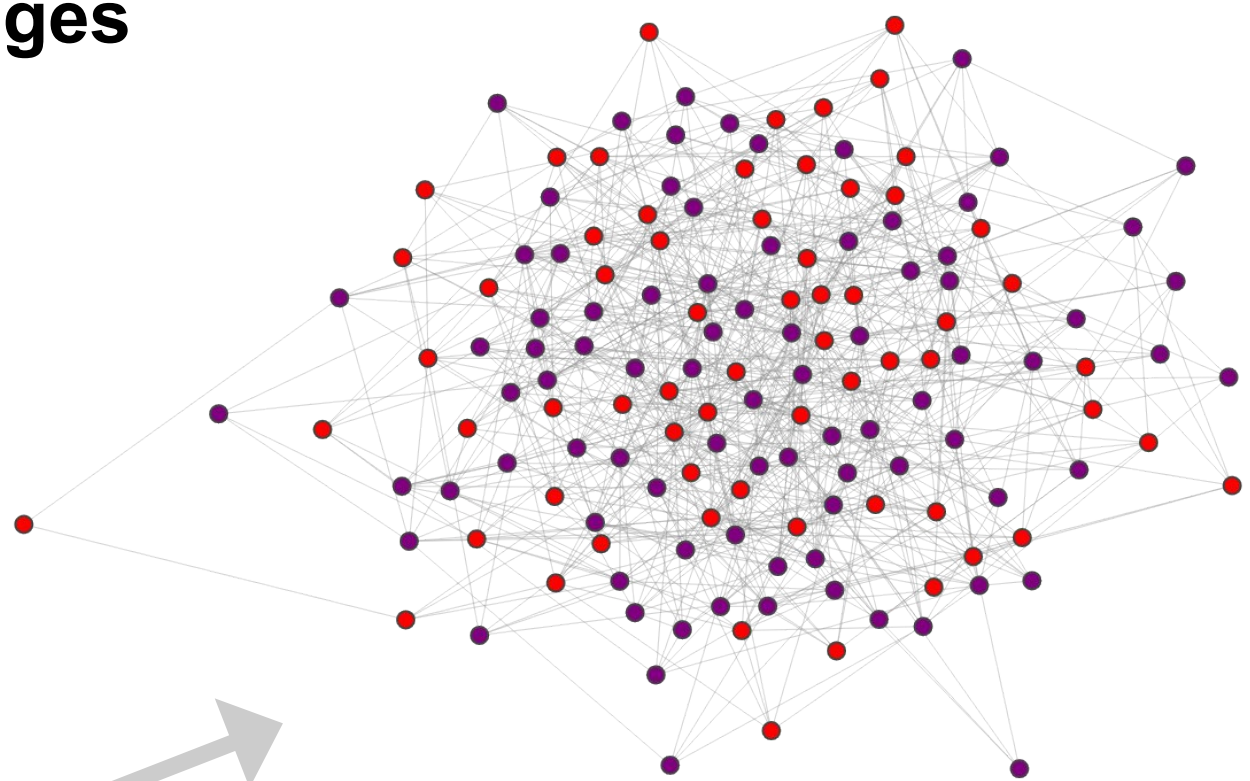
	Node	**Edge**	**Globally/Graph**
Social network	Name, age, job,...	Is friend, follows, sibling,...	Group of interest,...
Molecule	Atomic number,...	Bond order,...	Is a drug, energy,...
Citation	Article,...	Cited by,...	Research field,...
Particle physics	Particle,...	Decayed to,...	Experiment,...
Motion capture	Joints,...	Is connected to,...	Character,...
Natural language	Group of words,...	Refers to,...	Paragraph,...

The feature can be a number, a concept, ...

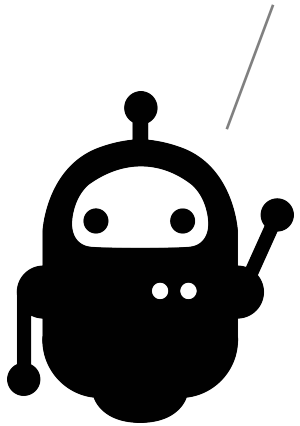
Formal definition

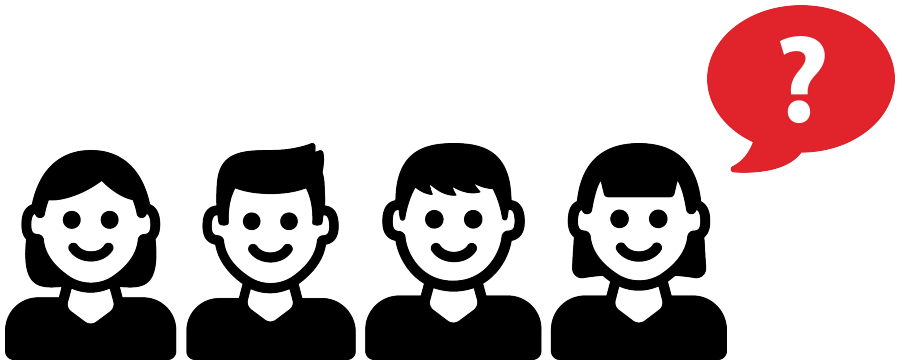
G = (V, E): a set of nodes and edges



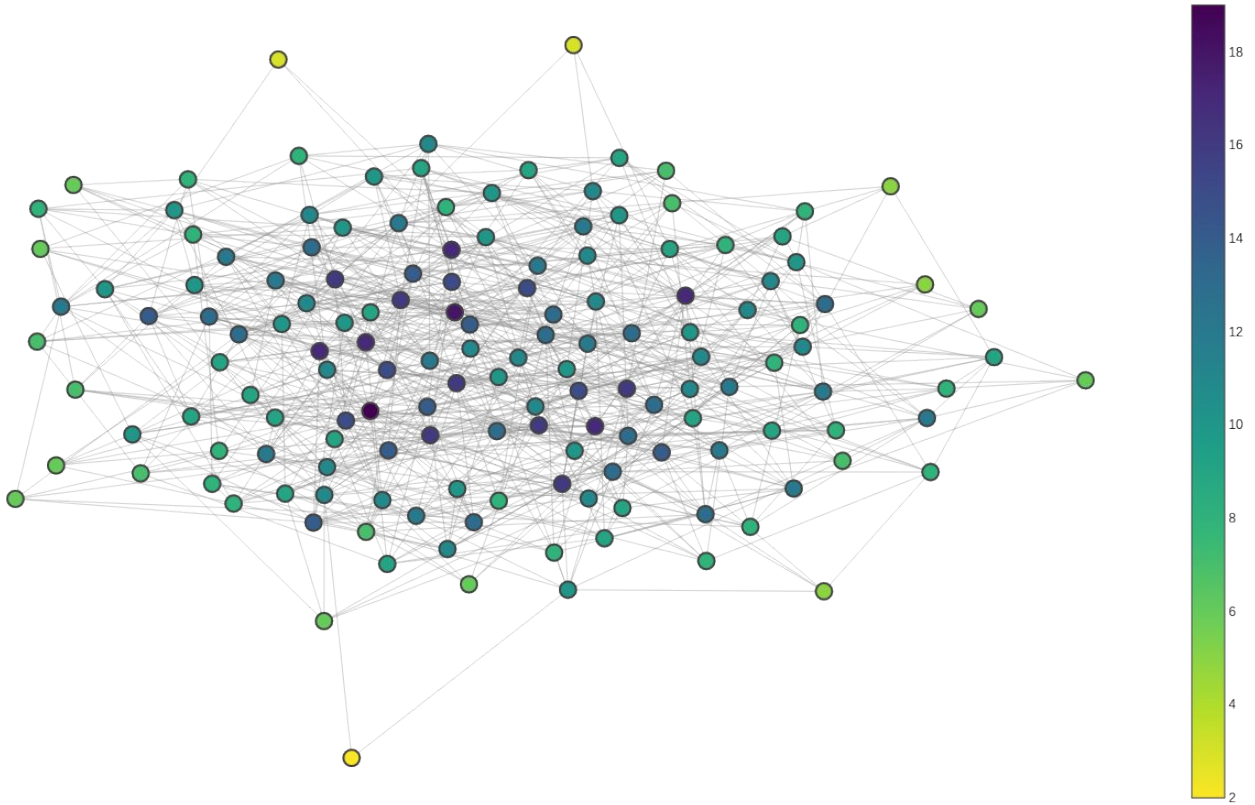


Any question?





Graph complexity



Number of neighbors

- The inner structure of a graph can vary a lot
- The number of edges/nodes might vary a lot from one graph to another
- One single graph can contain several thousands of nodes/edges
- ...

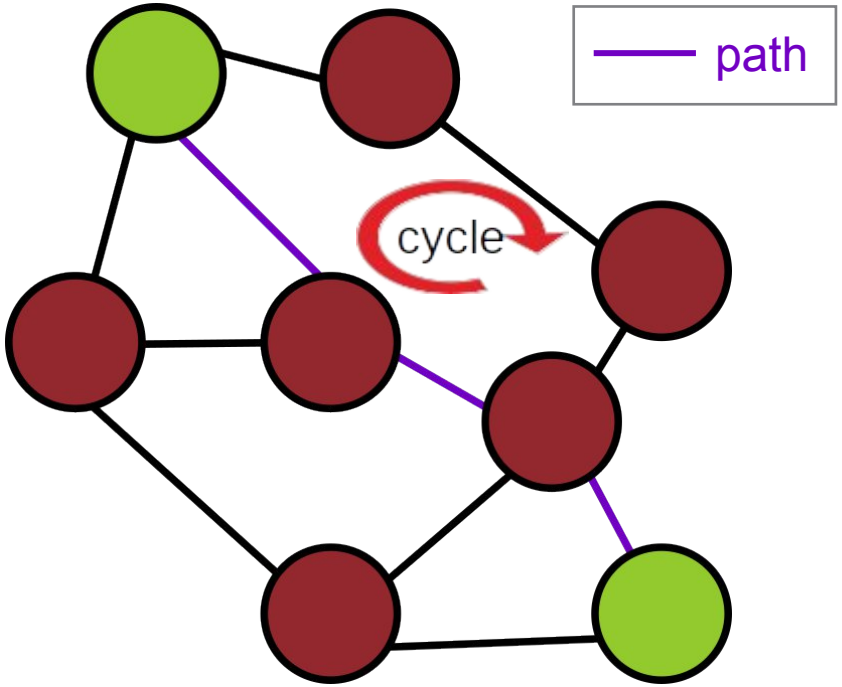
How do we quantify local or global structural patterns in a graph?

Graph complexity: Paths and cycles

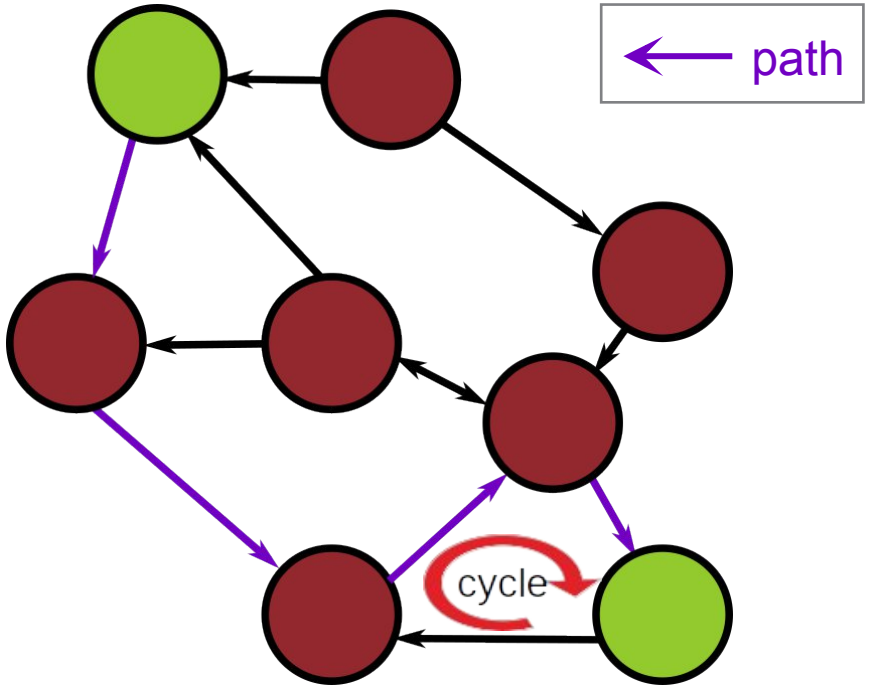
A **path** is a sequence of edges connecting 2 nodes

A **cycle** is a path in which the first and last vertices are equal

- **graph density**
- **graph redundancy**

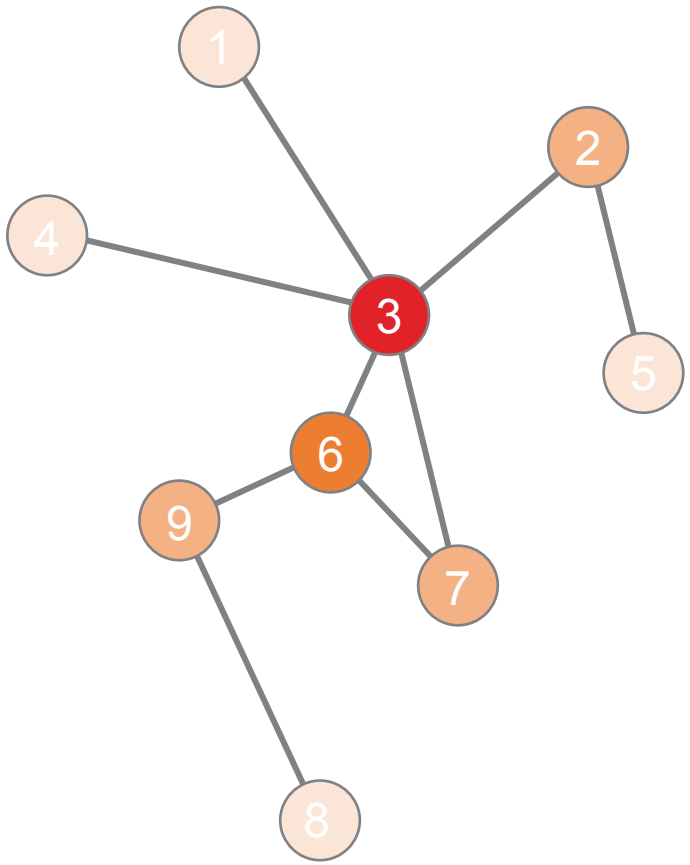


Undirected graph



Directed graph

Graph complexity: Node proximity and centrality



Node centrality

Node centrality

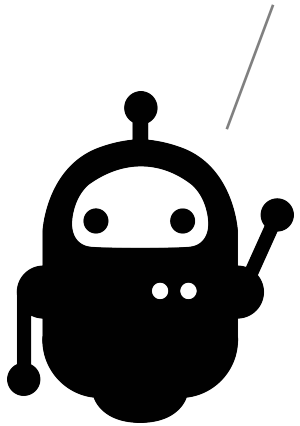
- How many paths goes through the node

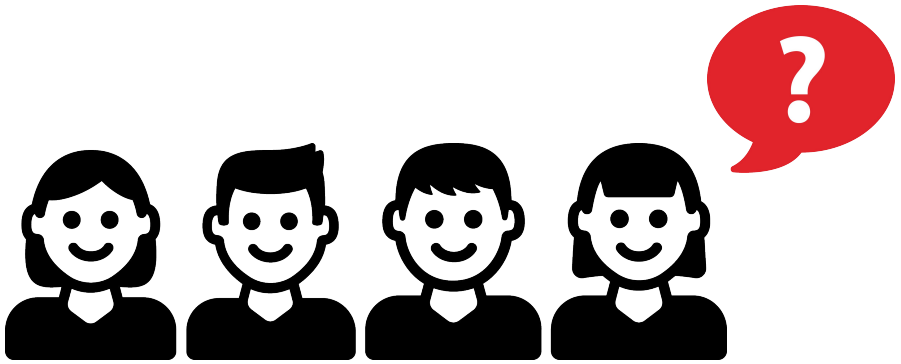
Node proximity measures
(connection strength between two nodes i and j)

- Shortest path (minimum number of edges going from i to j)
- « Heavier » path (maximum cumulated weight going from i to j)
- Probability to reach node j by performing a random walk starting from node i
- ...

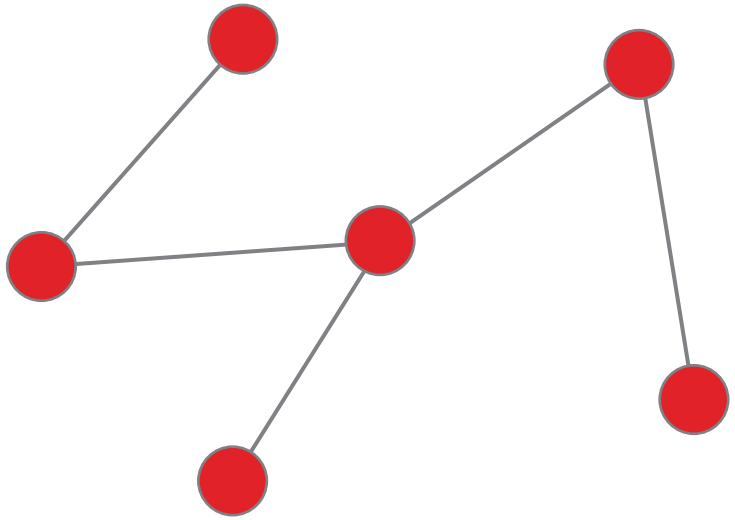
Useful for node and edge prediction

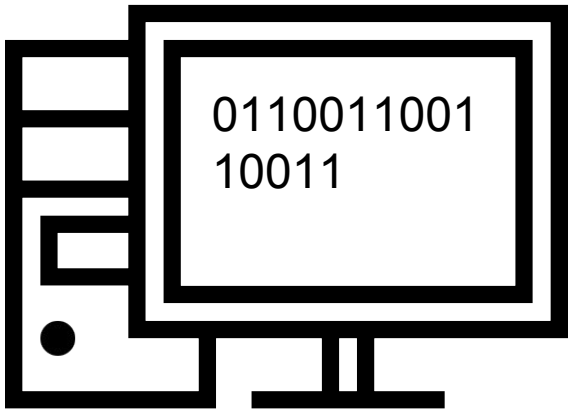
Any question?





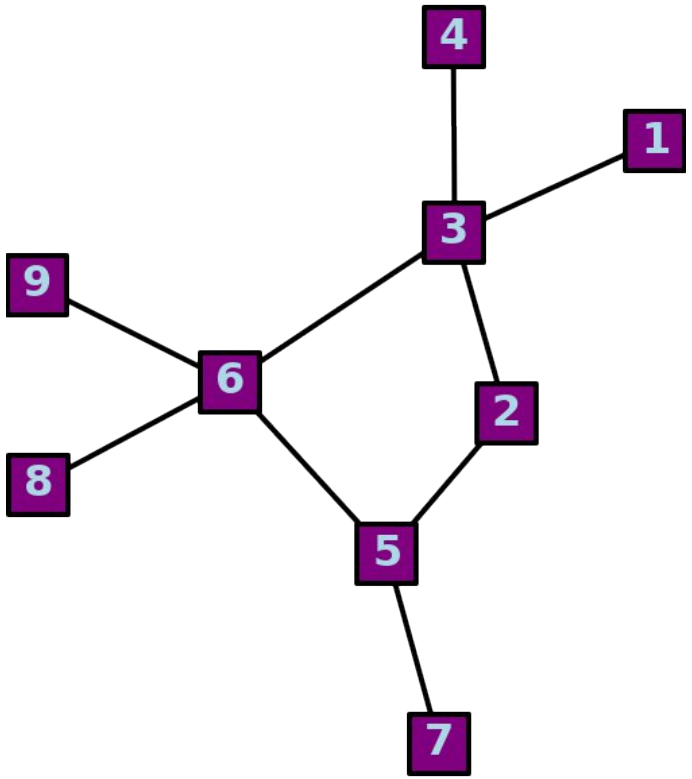
Graph representation





???

Graph representation

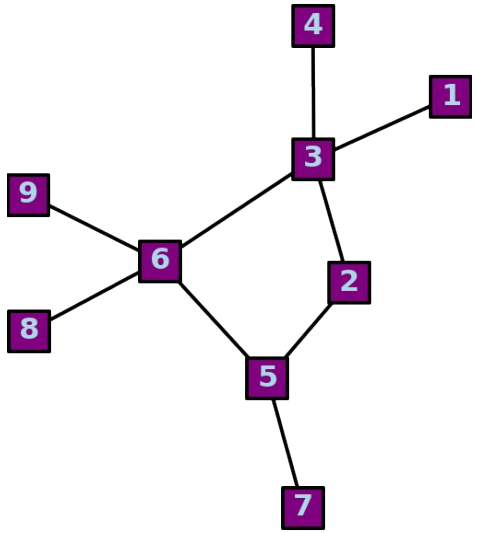


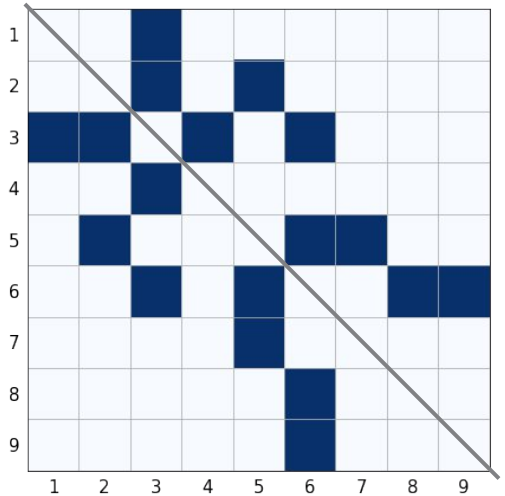
Random numbering of nodes

Graph representation

Adjacency matrix $W_{i,j} = \begin{cases} w_{i,j} \ if \ i \ is \ connected \ to \ j \\ 0 \ otherwise \end{cases}$

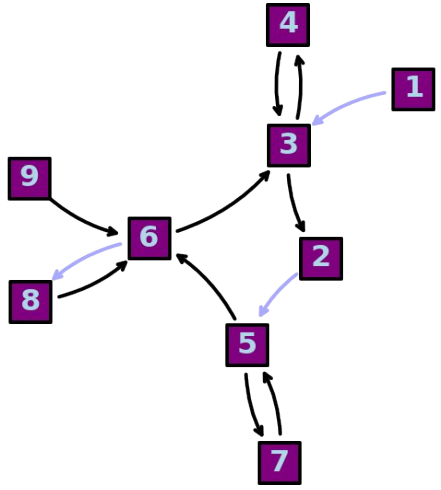
Undirected

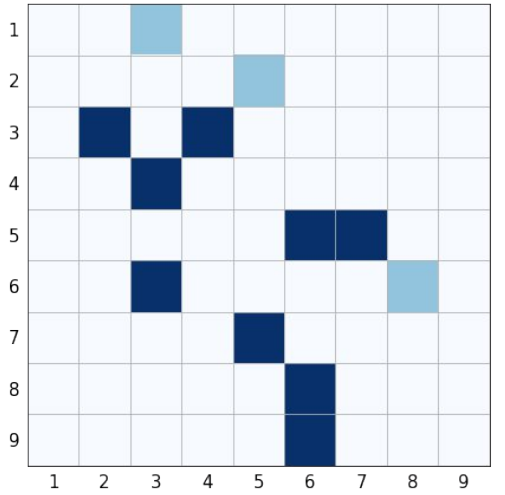




Symmetric

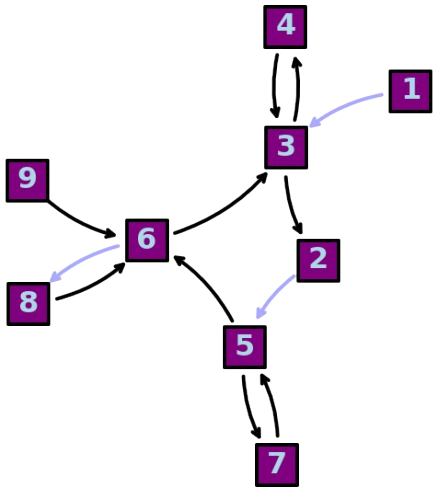
Directed

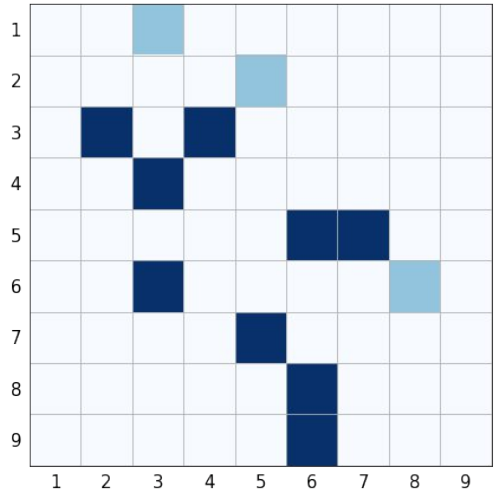




Graph representation

Adjacency list





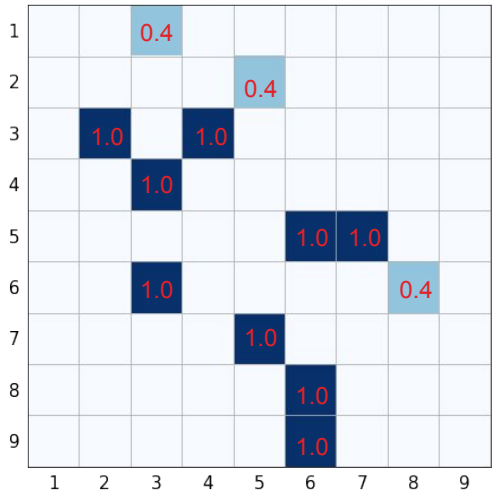
Adjacency list: [[1, 3],
[2, 5],
[3, 2], [3, 4],
[4, 3],
[5, 6], [5, 7],
[6, 3], [6, 8],
[7, 5],
[8, 6],
[9, 6]]

Edge weights: [0.4, 0.4, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 0.4, 1.0, 1.0, 1.0]

Graph representation

- **Edge weights** are stored either directly in the adjacency matrix, or in an independent tensor.

Adjacency matrix



Adjacency list: [[1, 3],
[2, 5],
[3, 2], [3, 4],
[4, 3],
[5, 6], [5, 7],
[6, 3], [6, 8],
[7, 5],
[8, 6],
[9, 6]]

Edge weights: [0.4, 0.4, 1.0,
1.0, 1.0, 1.0,
1.0, 1.0, 0.4,
1.0, 1.0, 1.0]

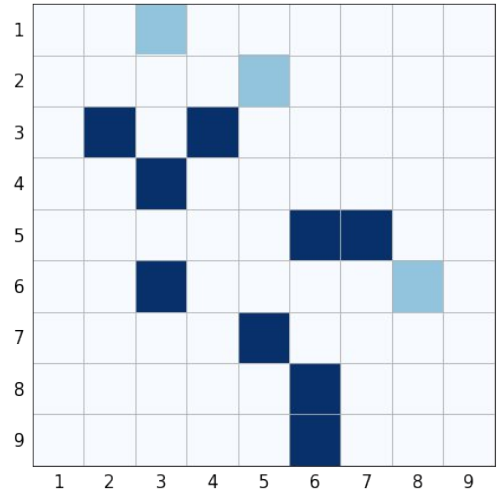
- **Features** on **nodes** and **graphs** will also be stored in independent tensors.

Node features: [4.1, 4.2, 6.4, 1.0, 1.0, 5.0, 1.7, 3.0, 5.0]

Graph feature: [8.0]

Graph representation

Adjacency matrix **VS** **Adjacency list**



Adjacency list: [[1, 3],
[2, 5],
[3, 2], [3, 4],
[4, 3],
[5, 6], [5, 7],
[6, 3], [6, 8],
[7, 5],
[8, 6],
[9, 6]]

Edge weights: [0.4, 0.4, 1.0,
1.0, 1.0, 1.0,
1.0, 1.0, 0.4,
1.0, 1.0, 1.0]

Storage space	$O(V^2)$ → lot of space	$O(E)$ → less space
Storage efficiency	Might be sparse	Optimal
Does edge [i,j] exist?	Check if $W(i,j) \neq 0$ → $O(1)$	Cover the list until finding the edge → $O(E)$
Finding i's neighbours	Read the i^{th} line → $O(V)$	Cover the whole list → $O(E)$

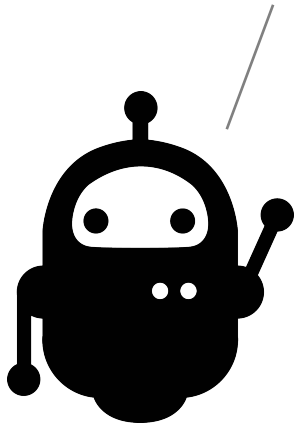
V = number of vertices
E = number of edges

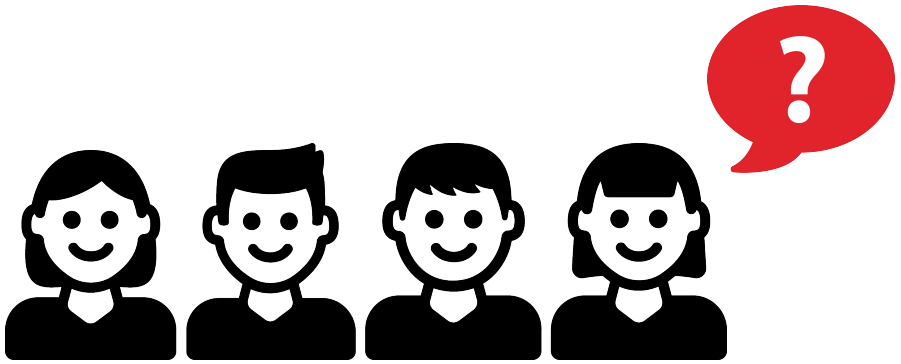
Graph representation

- Useful matrices

Adjacency	**W**	Edge weights
Degree	**D**	Number of neighbours per node (diagonal matrix)
Laplacian	**L**	**D - W** (~ diffusion matrix)

Any question?





1 **Graphs are everywhere**

- Complex data structures
- Basics of graph theory

2 **Learning on Graphs**

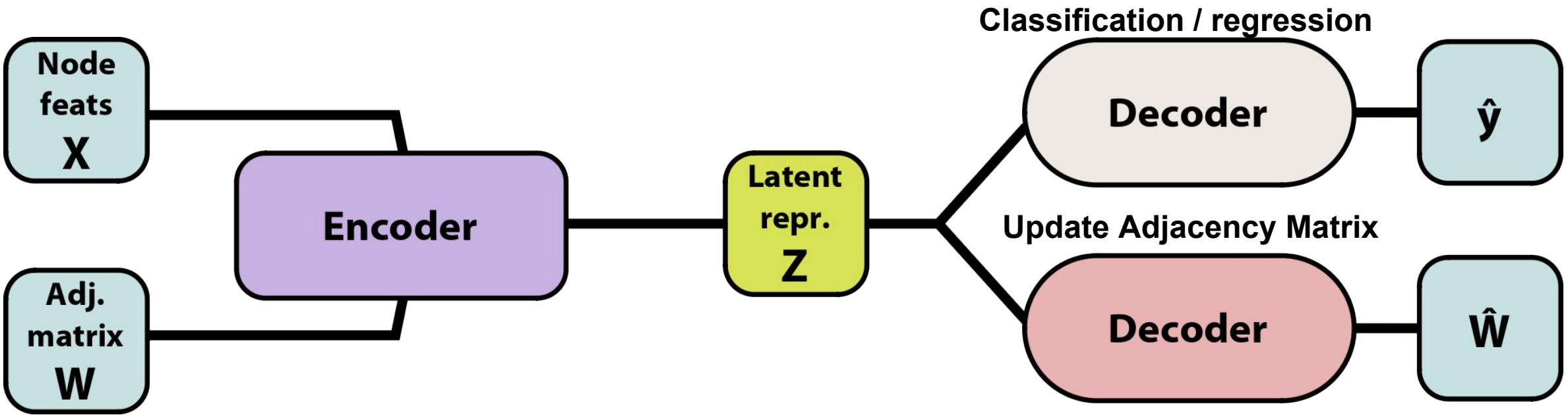
- Graph embedding
- Transductive and inductive learning
- Tasks on graph learning

3 **A few examples**

- Taxonomy of methods
- Graph convolution
- Message passing
- Graph Transformer

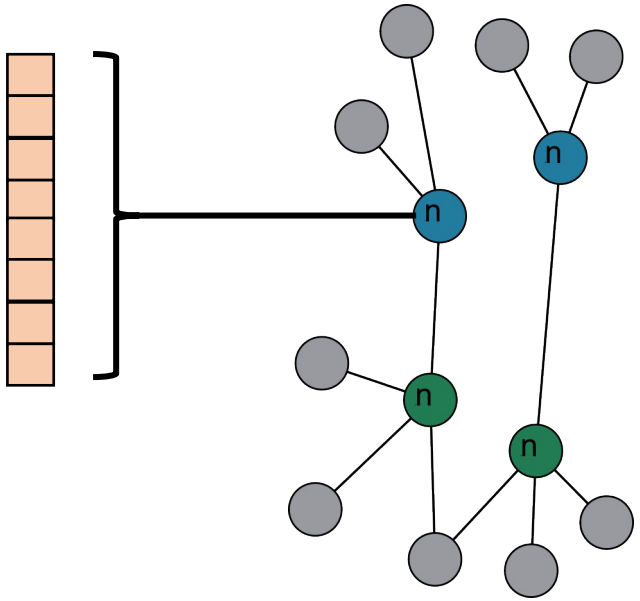
Graph Embedding

We need to find a representation of the graph that is processable



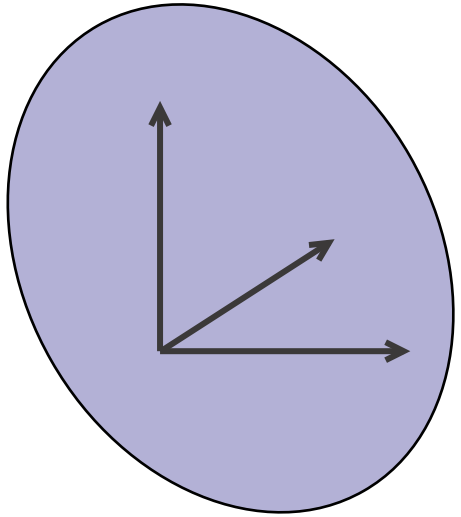
Graph Embedding: Representing nodes

Node Features



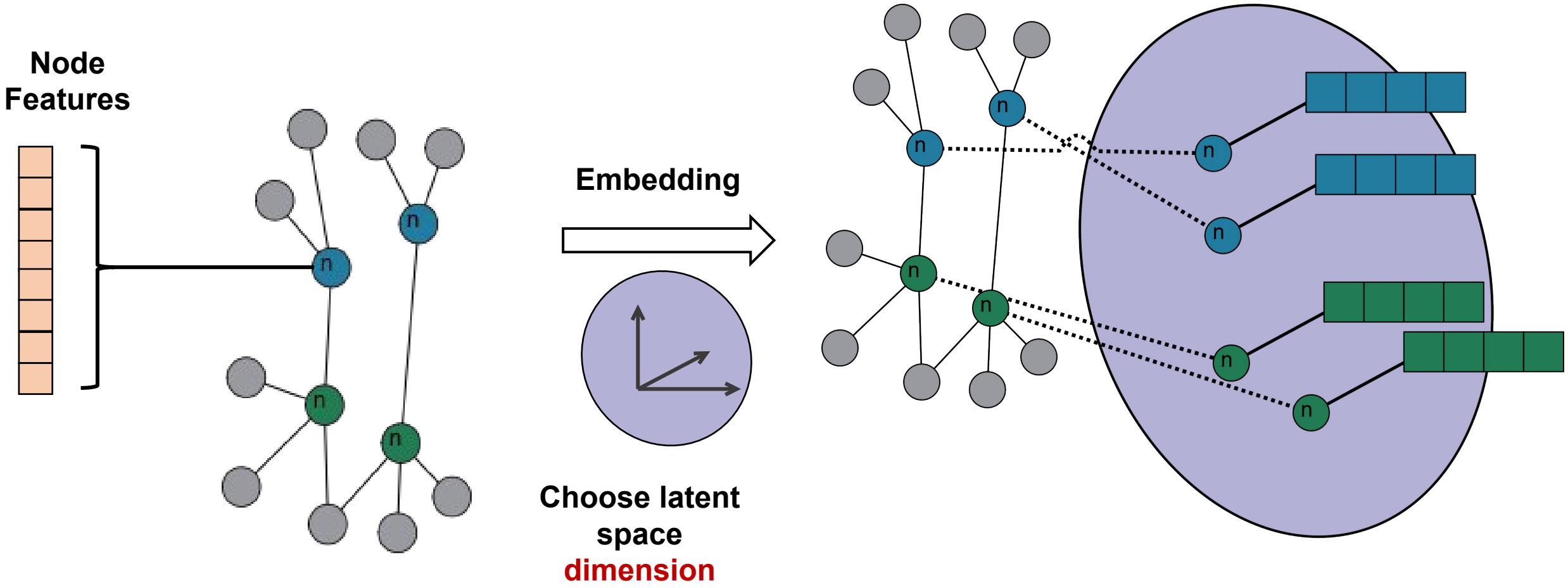
Embedding

Choose latent space dimension → **Hyperparameter**



Node Embedding: Representing features

Each node is represented in the latent space

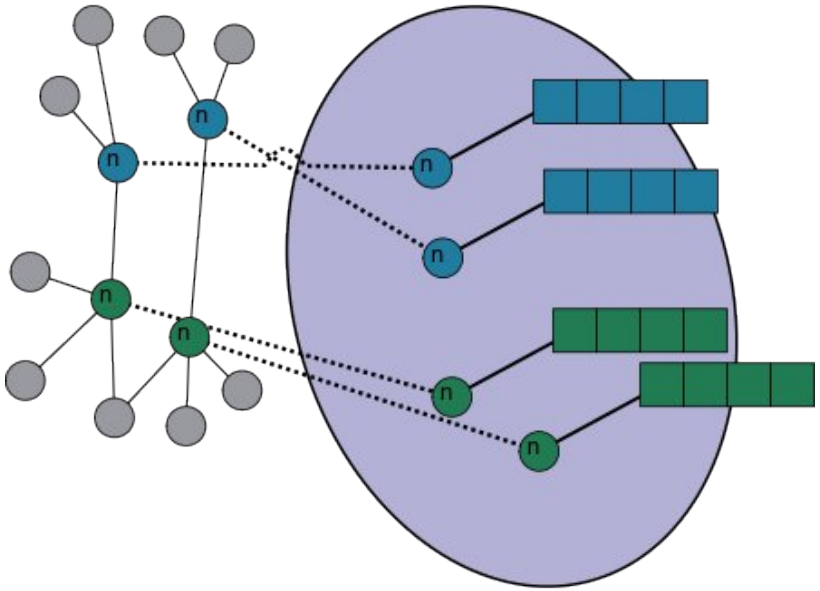


Node embedding: right representation for task

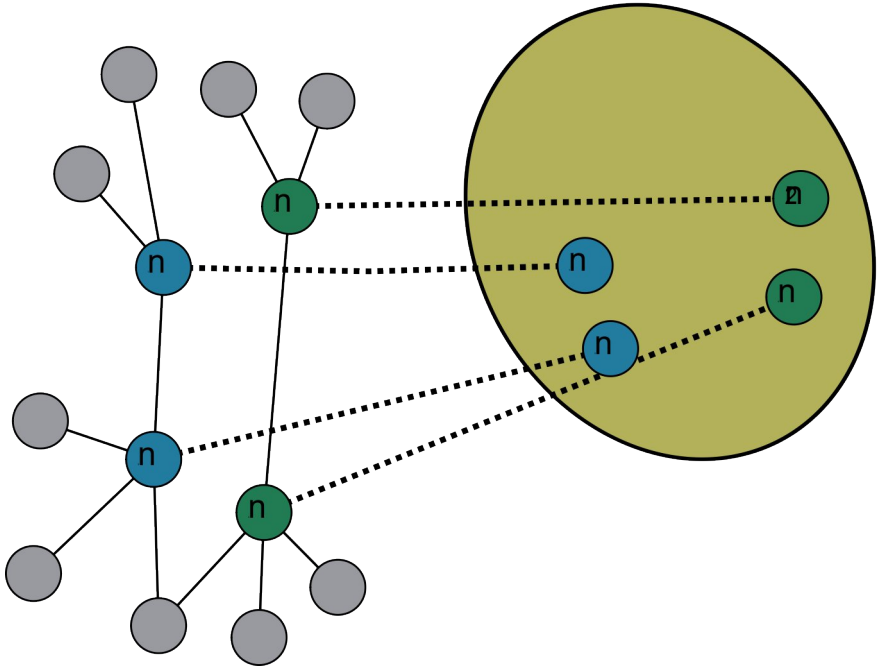
Different tasks = different latent space

⇾ Learnable

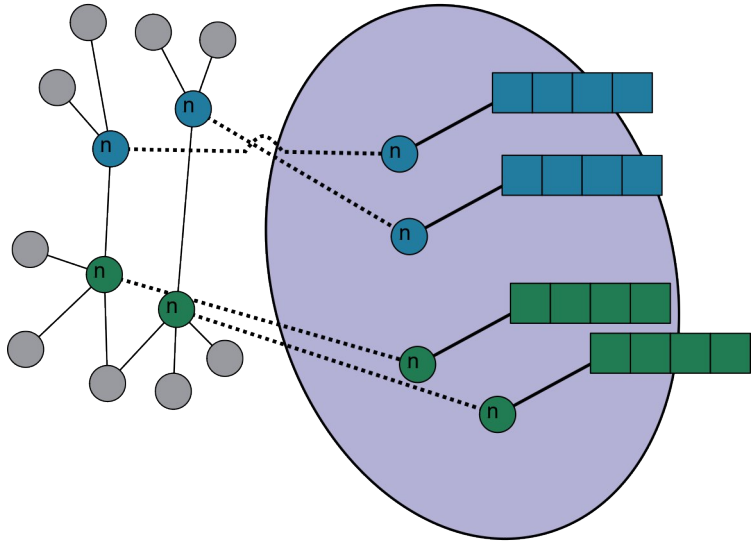
Example 1 : Nodes with similar environment close in latent space

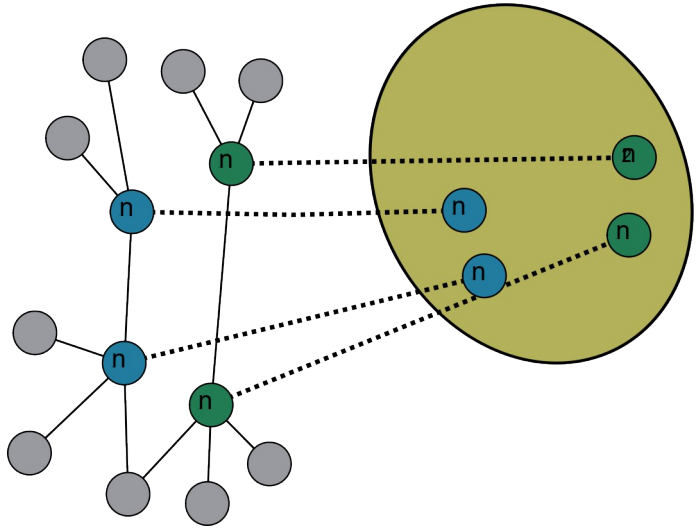


Example 2 : Nodes close in graph close in latent space

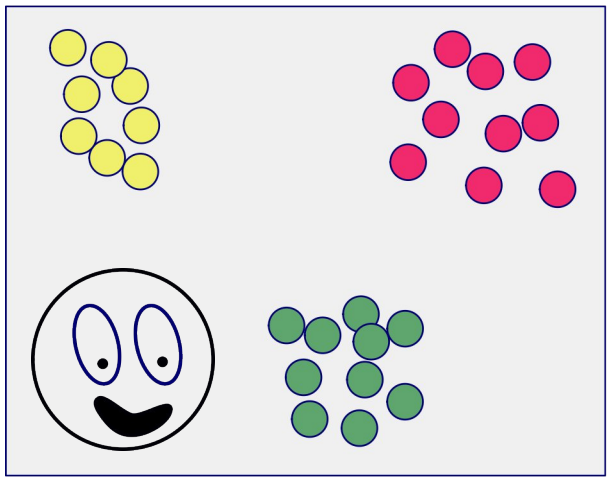


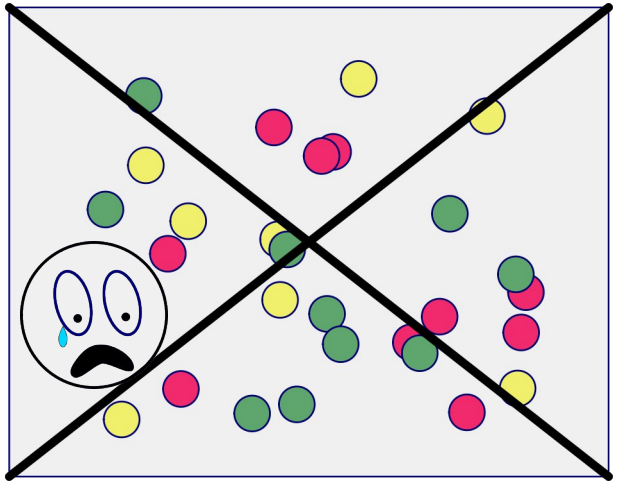
Node embedding: Summary





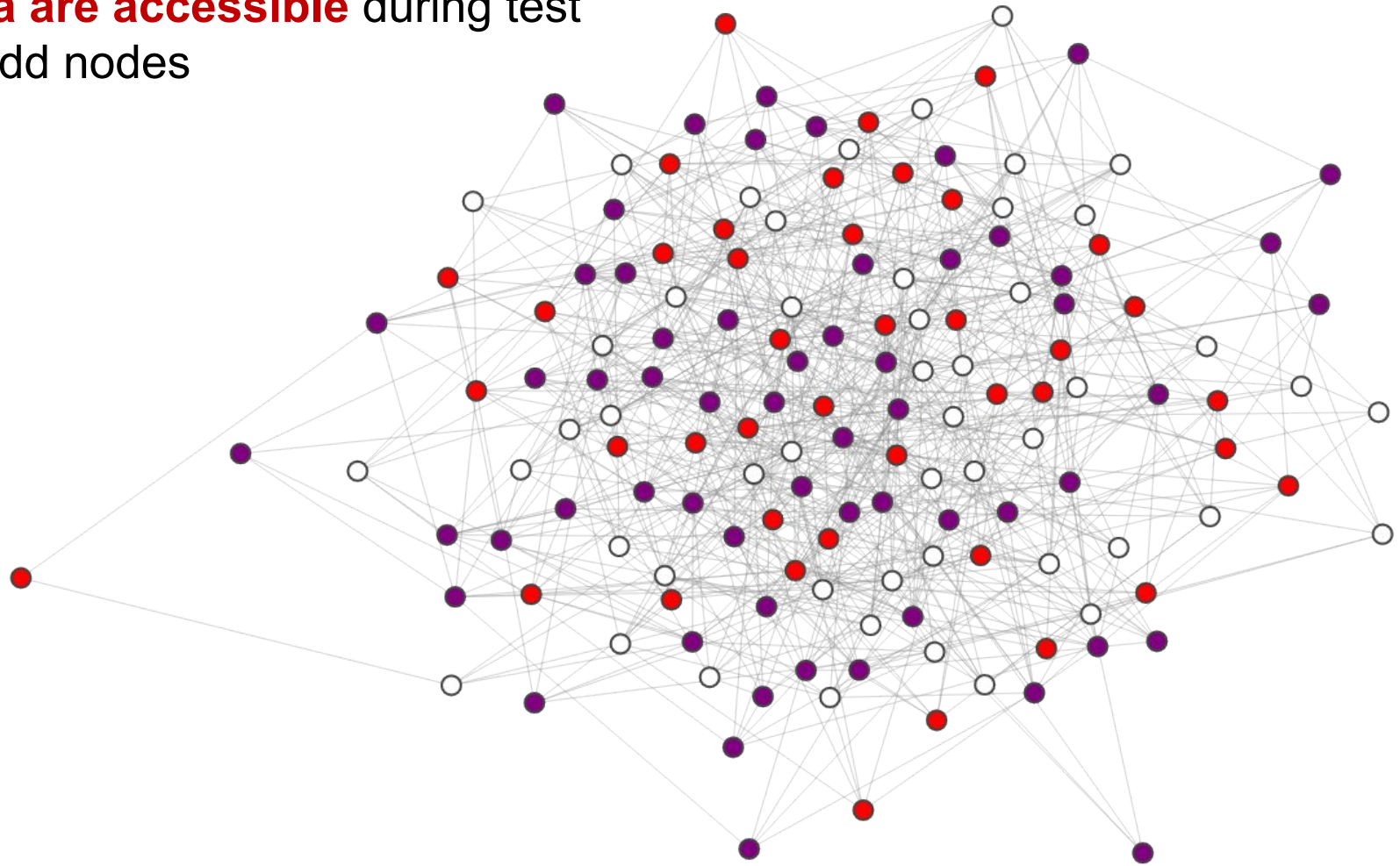
- Features stored in nodes/edges/graphs are not easily processed.
- We transform the features into a vector in the latent space (**Dimension is a hyperparameter**).
- The embedding has to be suited for the task → **Learnable**.





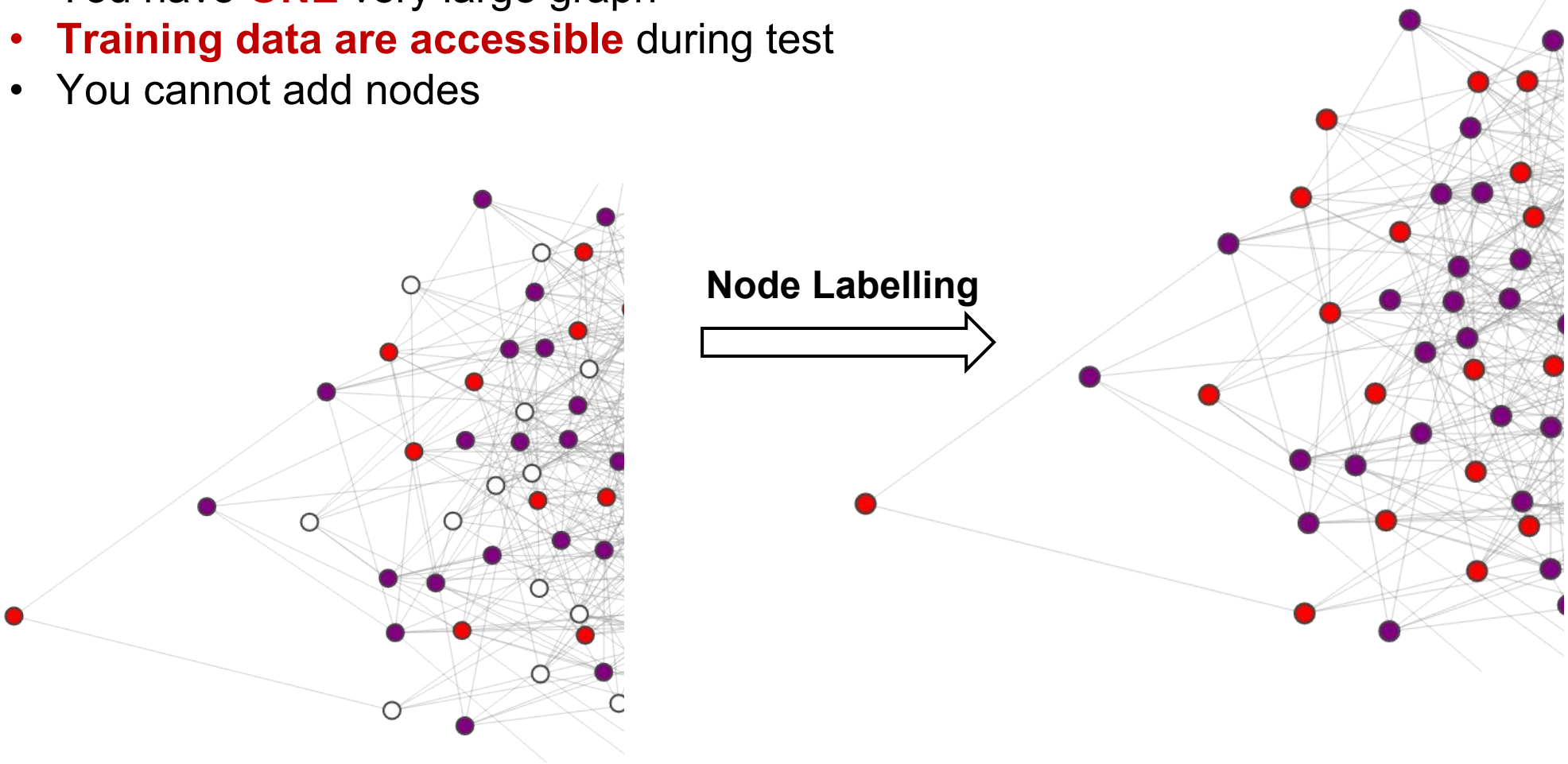
Transductive Learning

- You have **ONE** very large graph
- **Training data are accessible** during test
- You cannot add nodes



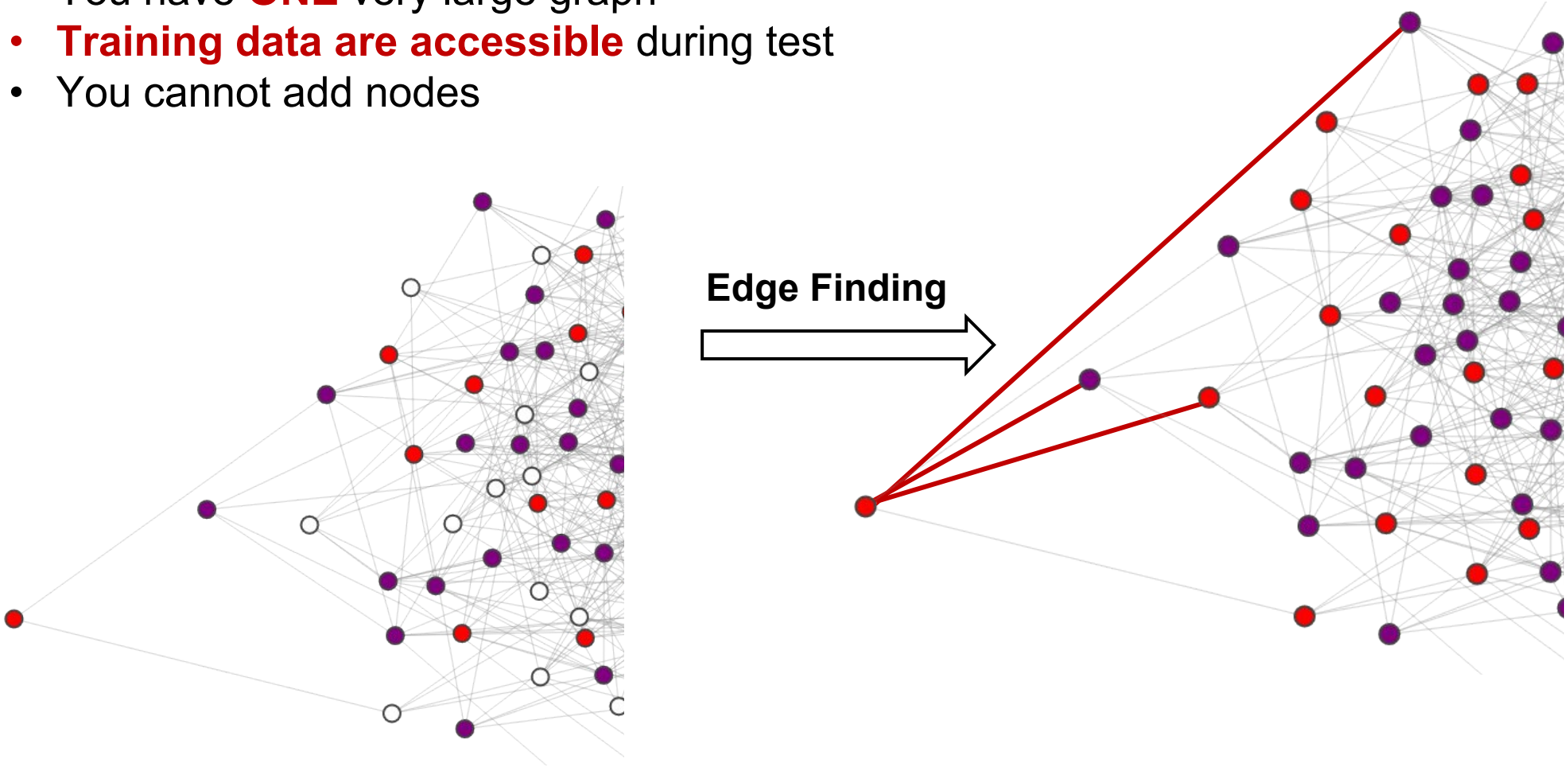
Transductive Learning : Node labelling

- You have **ONE** very large graph
- **Training data are accessible** during test
- You cannot add nodes



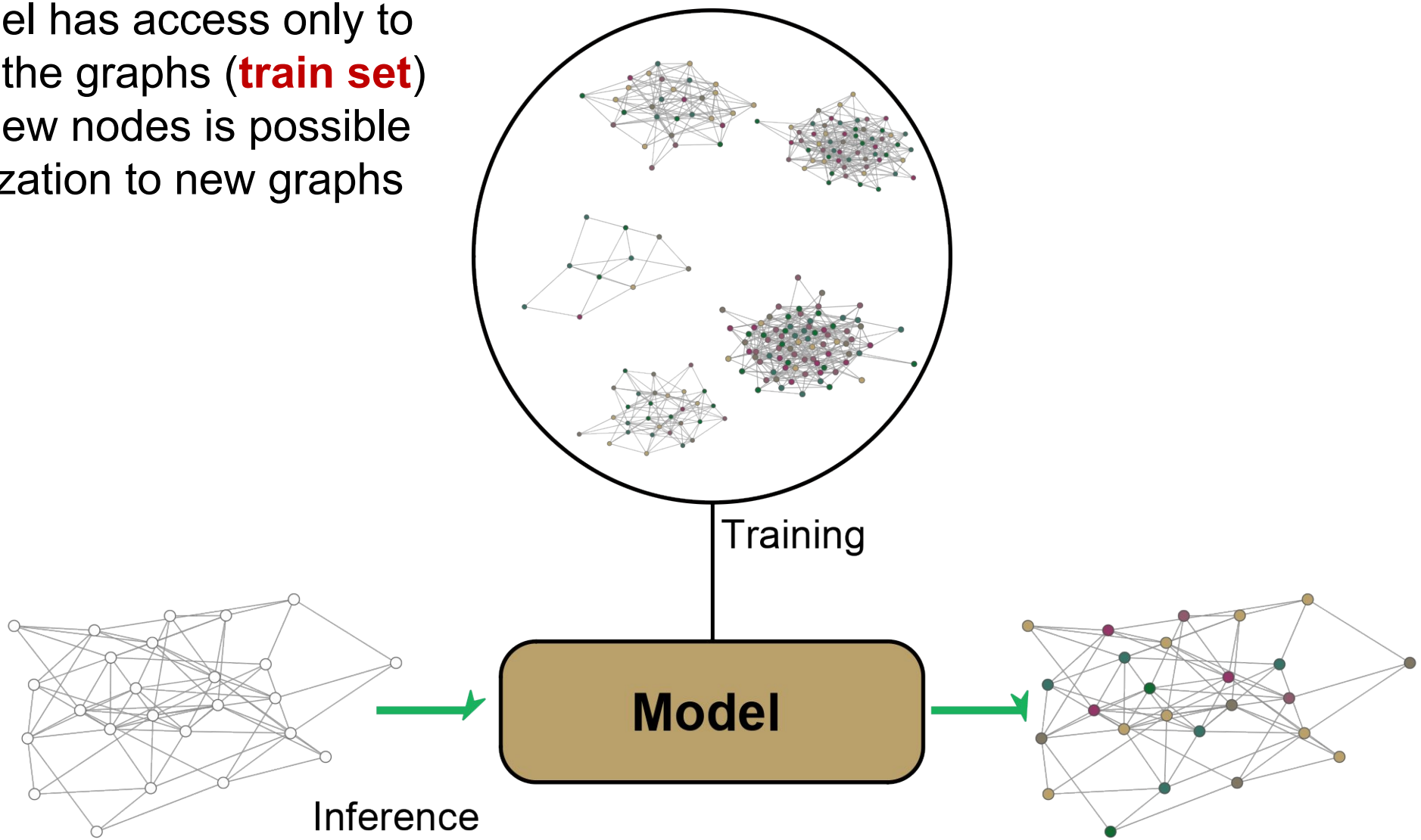
Transductive Learning : Edge finding

- You have **ONE** very large graph
- **Training data are accessible** during test
- You cannot add nodes

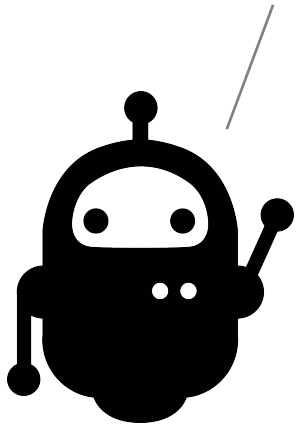


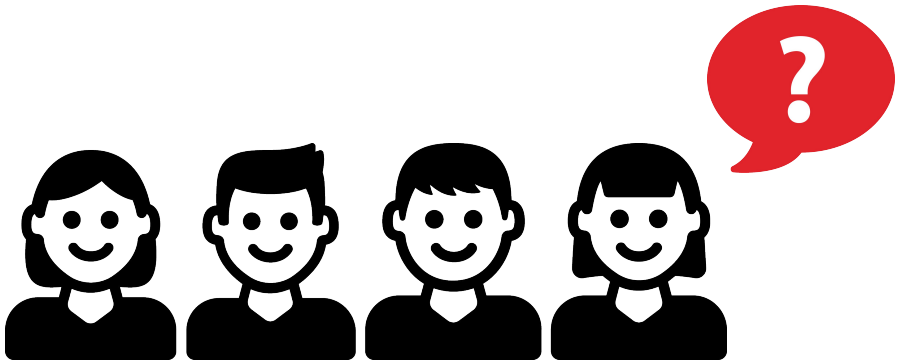
Inductive Learning

- The model has access only to a part of the graphs (**train set**)
- Adding new nodes is possible
- Generalization to new graphs



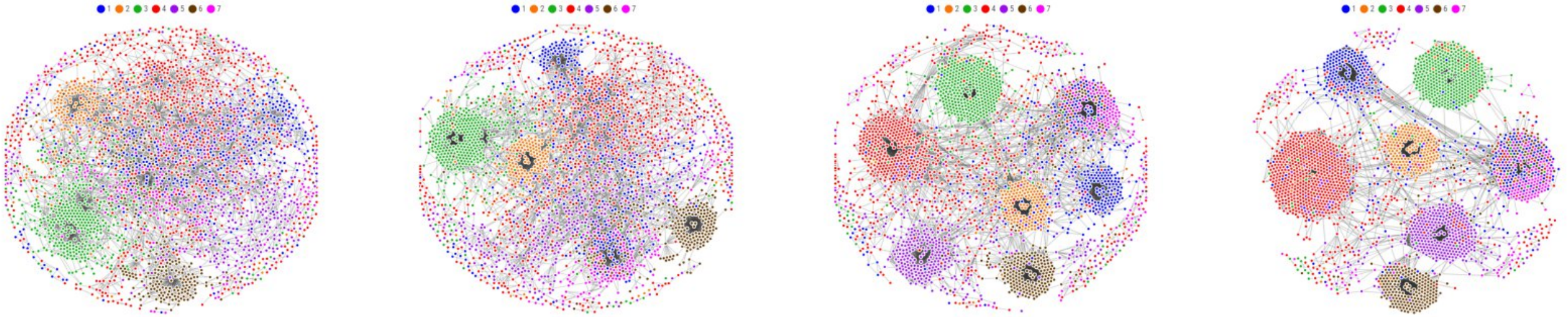
Any question?





Tasks on nodes: Labelling

Node clustering (example on CORA dataset)

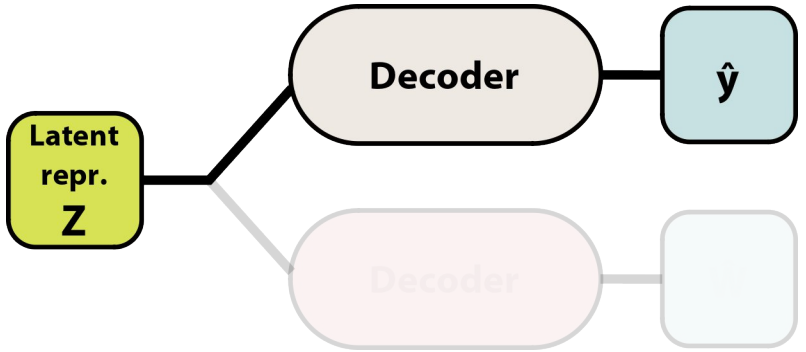


(a) Epoch 0 (b) Epoch 40 (c) Epoch 80 (d) Epoch 120

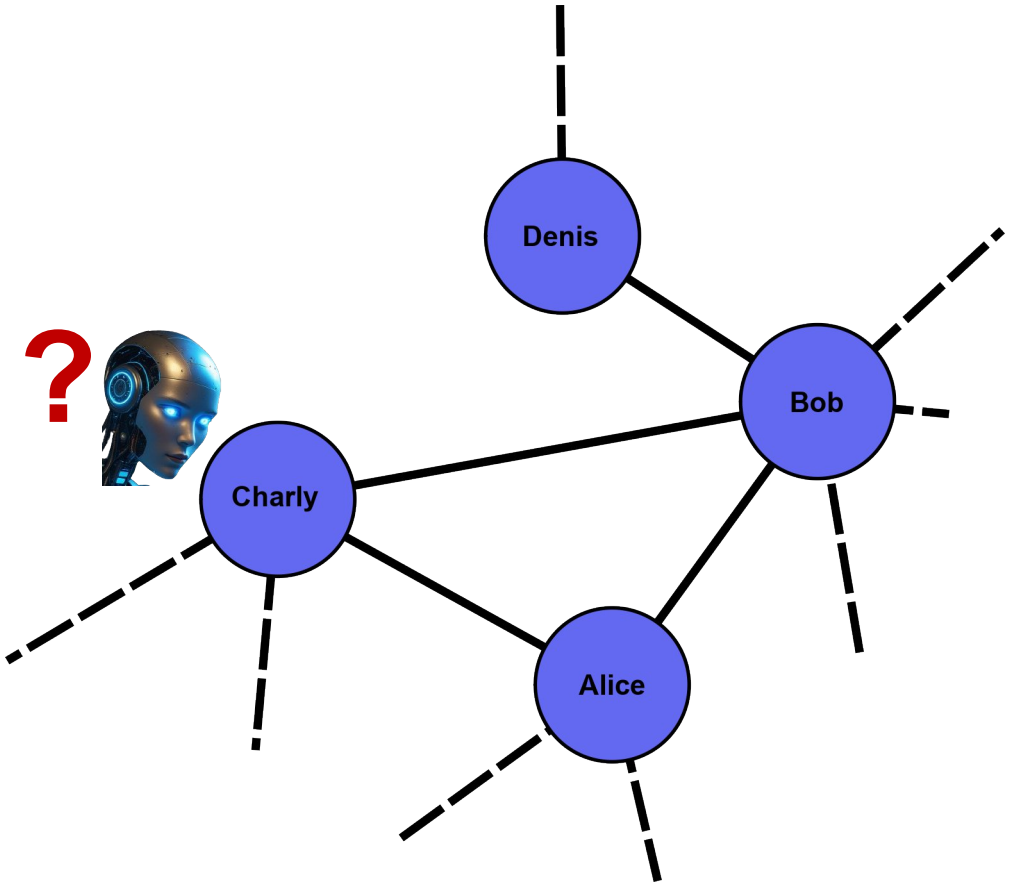
Cora dataset: McCallum, A. K.; Nigam, K.; Rennie, J.; Seymore, K. Automating the Construction of Internet Portals with Machine Learning. *Information Retrieval* **2000**, *3* (2), 127–163. .

Model: Mrabah, N.; Bouguessa, M.; Touati, M. F.; Ksantini, R. Rethinking Graph Auto-Encoder Models for Attributed Graph Clustering (Extended Abstract). In *2023 IEEE 39th International Conference on Data Engineering (ICDE)*; 2023; pp 3891–3892. .

Tasks on nodes: Labelling



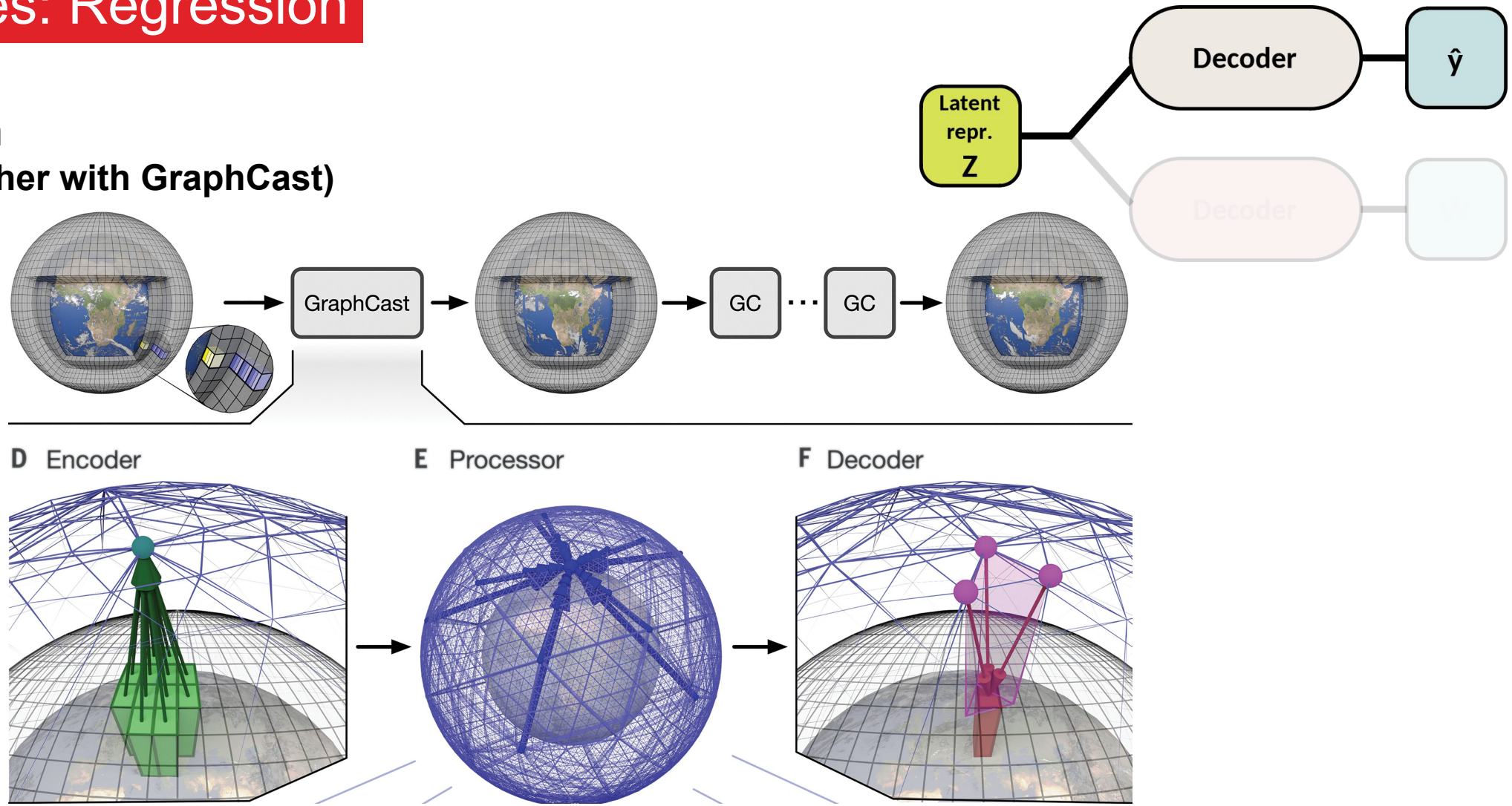
Node labelling (eg. bot detection on social networks)



Example: Huang, H.; Tian, H.; Zheng, X.; Zhang, X.; Zeng, D. D.; Wang, F.-Y. CGNN: A Compatibility-Aware Graph Neural Network for Social Media Bot Detection. *IEEE Transactions on Computational Social Systems* **2024**, *11* (5), 6528–6543. .

Tasks on nodes: Regression

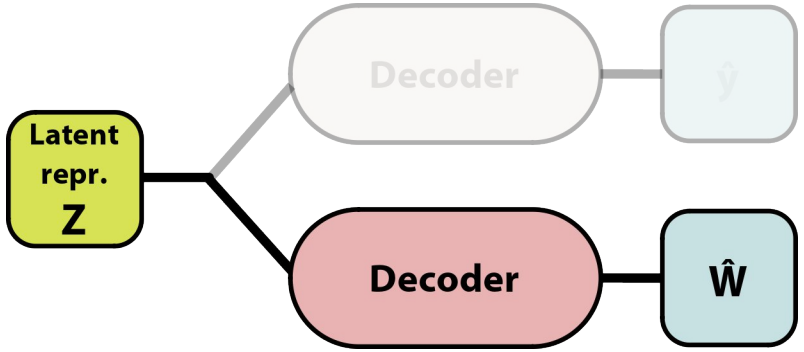
Node regression
(eg. predicting weather with GraphCast)

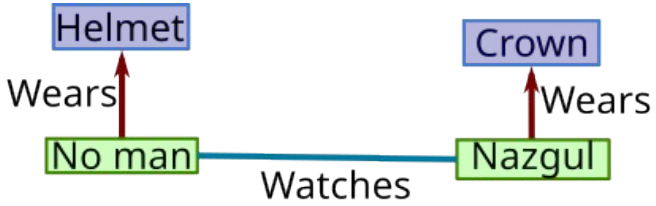


Example: Lam, R.; Sanchez-Gonzalez, A.; Willson, M.; Wirnsberger, P.; Fortunato, M.; Alet, F.; Ravuri, S.; Ewalds, T.; Eaton-Rosen, Z.; Hu, W.; Merose, A.; Hoyer, S.; Holland, G.; Vinyals, O.; Stott, J.; Pritzel, A.; Mohamed, S.; Battaglia, P. Learning Skillful Medium-Range Global Weather Forecasting. *Science* **2023**. .

Tasks on edges

Find relationship between objects
(eg. scene graph generation)





- Build the graph with semantic segmentation
- Find the relation between elements

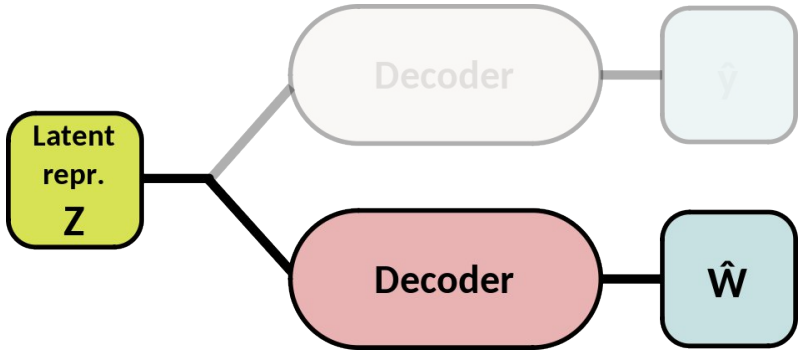
6

Convolutional models
CNN

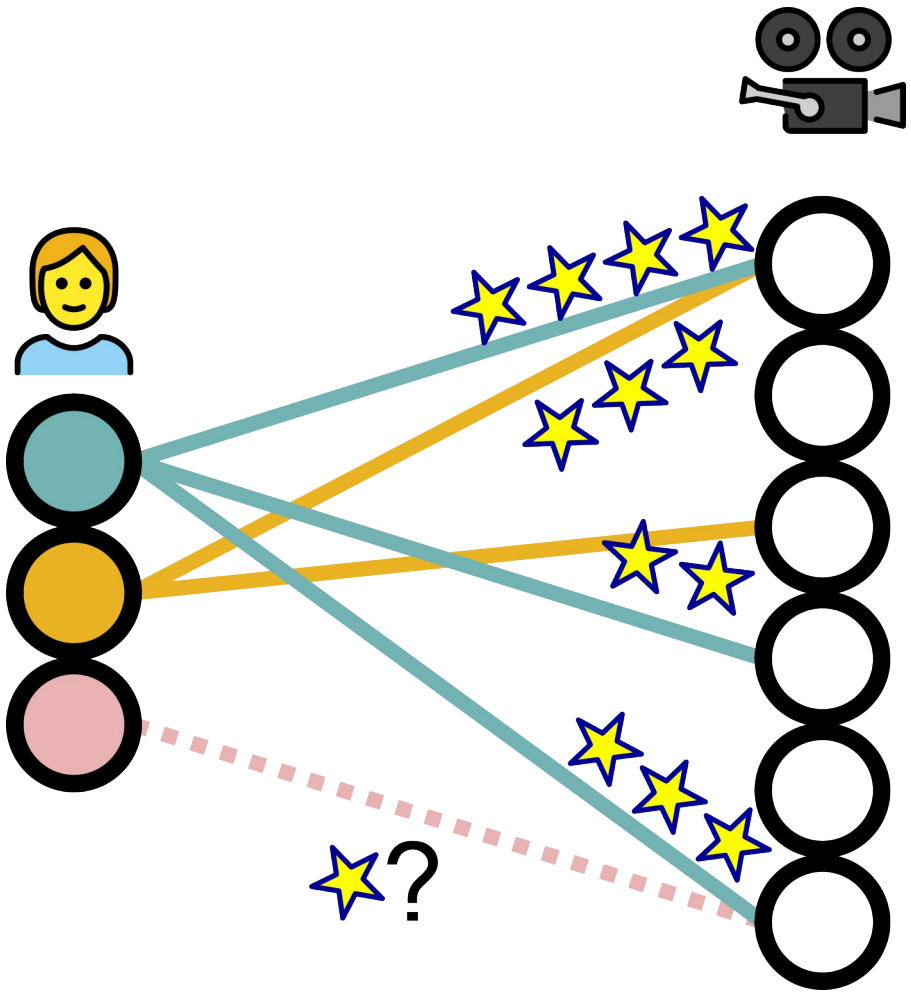
Example : Zhu, G.; Zhang, L.; Jiang, Y.; Dang, Y.; Hou, H.; Shen, P.; Feng, M.; Zhao, X.; Miao, Q.; Shah, S. A. A.; Bennamoun, M. Scene Graph Generation: A Comprehensive Survey. arXiv June 22, 2022. .

Tasks on edges

Edge prediction/regression
(eg. movie recommendation)



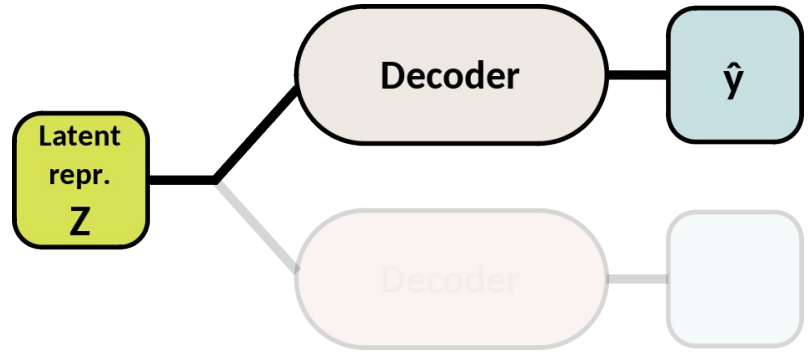
- **Transductive** learning
- Bipartite graph
  (2 different kinds of nodes)
- Try to find the **weight** of an edge



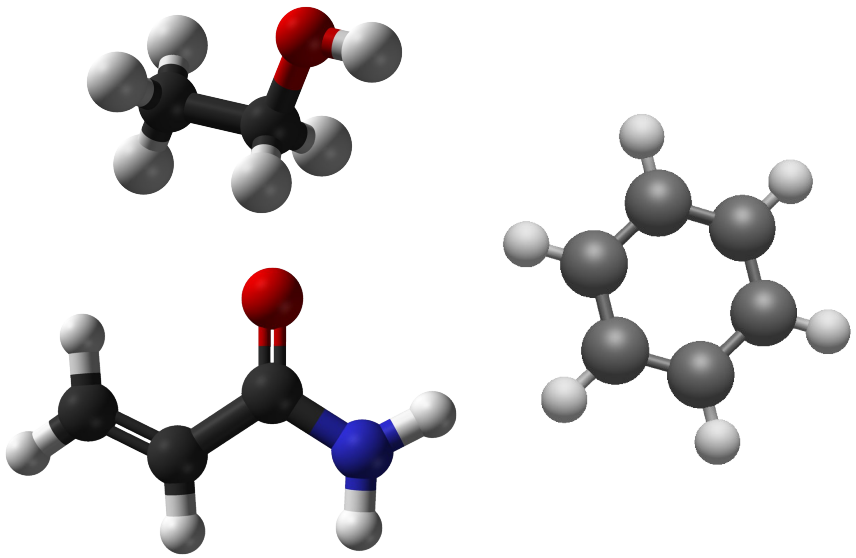
Example :
Zhang, M.; Chen, Y. Inductive Matrix Completion Based on Graph Neural Networks. arXiv February 16, 2020.

Tasks on graphs

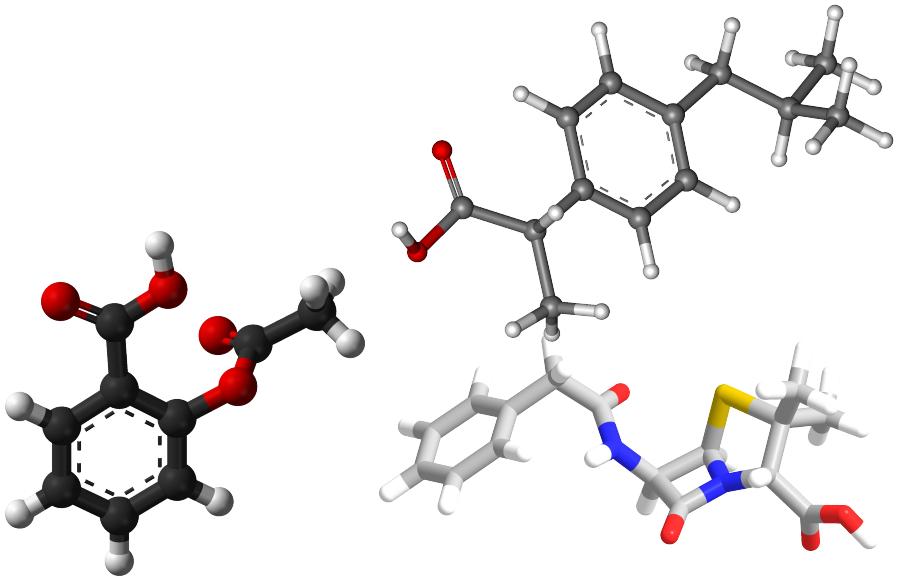
Graph classification
(eg. carcinogen / drug)



Carcinogens

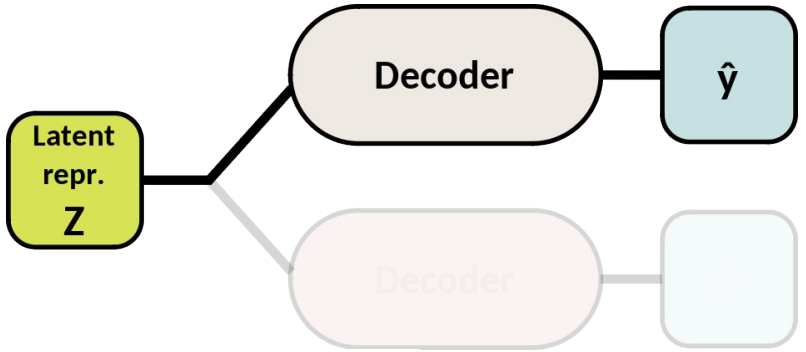


Drug



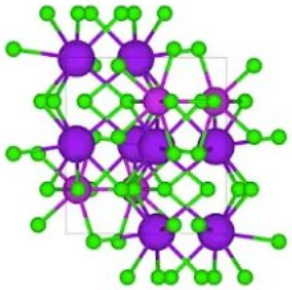
Example : Wang, C.; Kumar, G. A.; Rajapakse, J. C. Drug Discovery and Mechanism Prediction with Explainable Graph Neural Networks. *Sci Rep* **2025**, *15* (1), 179. .

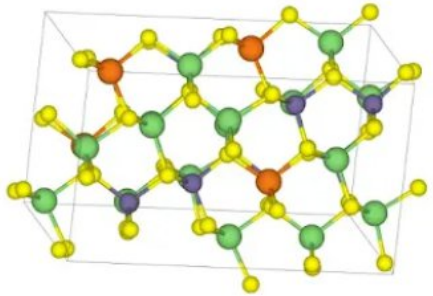
Tasks on graphs

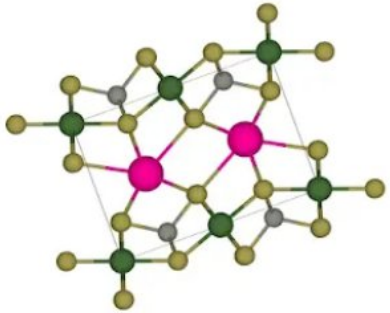


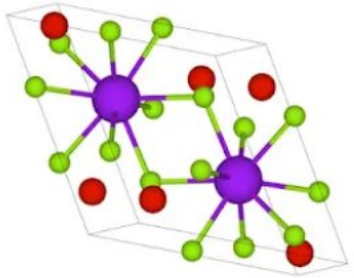
Graph regression
(eg. crystal structure stability with GNoME)

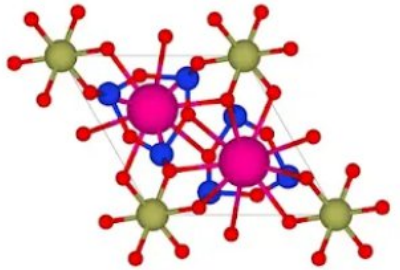
- Predict stability from structure
- Generate new structures
- Active learning

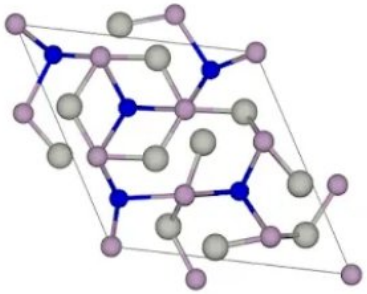

K_2BiCl_5


$Li_4MgGe_2S_7$


Mo_5GeB_2

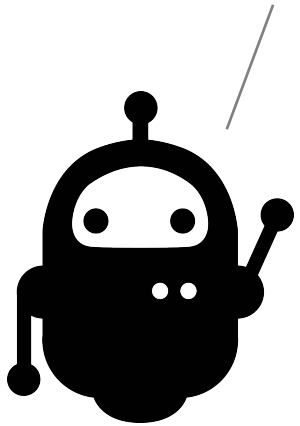

KV_3SE_3

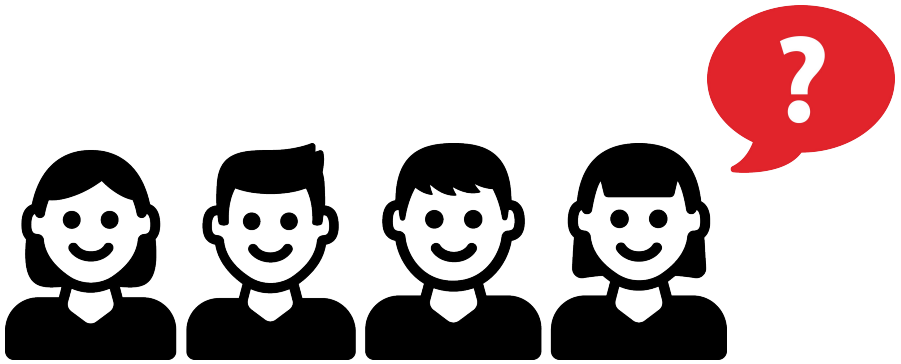

$Rb_2HfSi_3O_9$


$Tm_5Pd_9P_7$

Example: Merchant, A.; Batzner, S.; Schoenholz, S. S.; Aykol, M.; Cheon, G.; Cubuk, E. D. Scaling Deep Learning for Materials Discovery. *Nature* **2023**, *624* (7990), 80–85. .

Any question?





1 Graphs are everywhere

- Complex data structures
- Basics of graph theory

2 Learning on Graphs

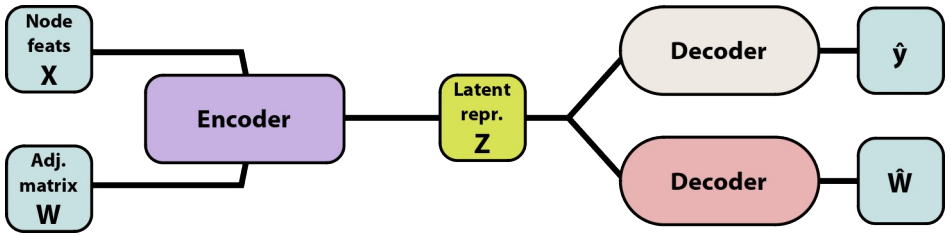
- Graph embedding
- Transductive and inductive learning
- Tasks on graph learning

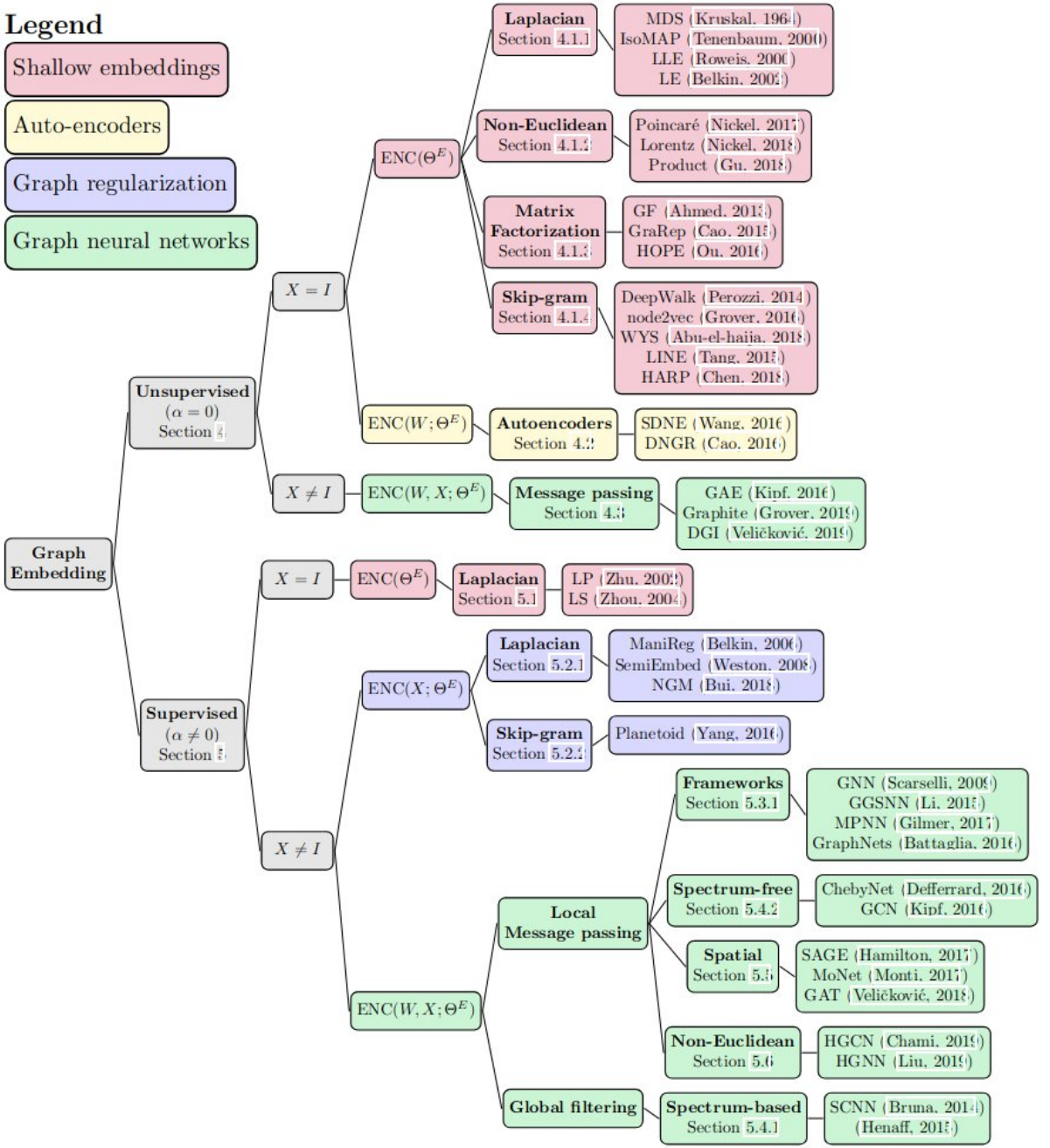
3 A few examples

- Taxonomy of methods
- Graph convolution
- Message passing
- Graph Transformer

Taxonomy of methods

There are many different flavors of methods for learning on graphs

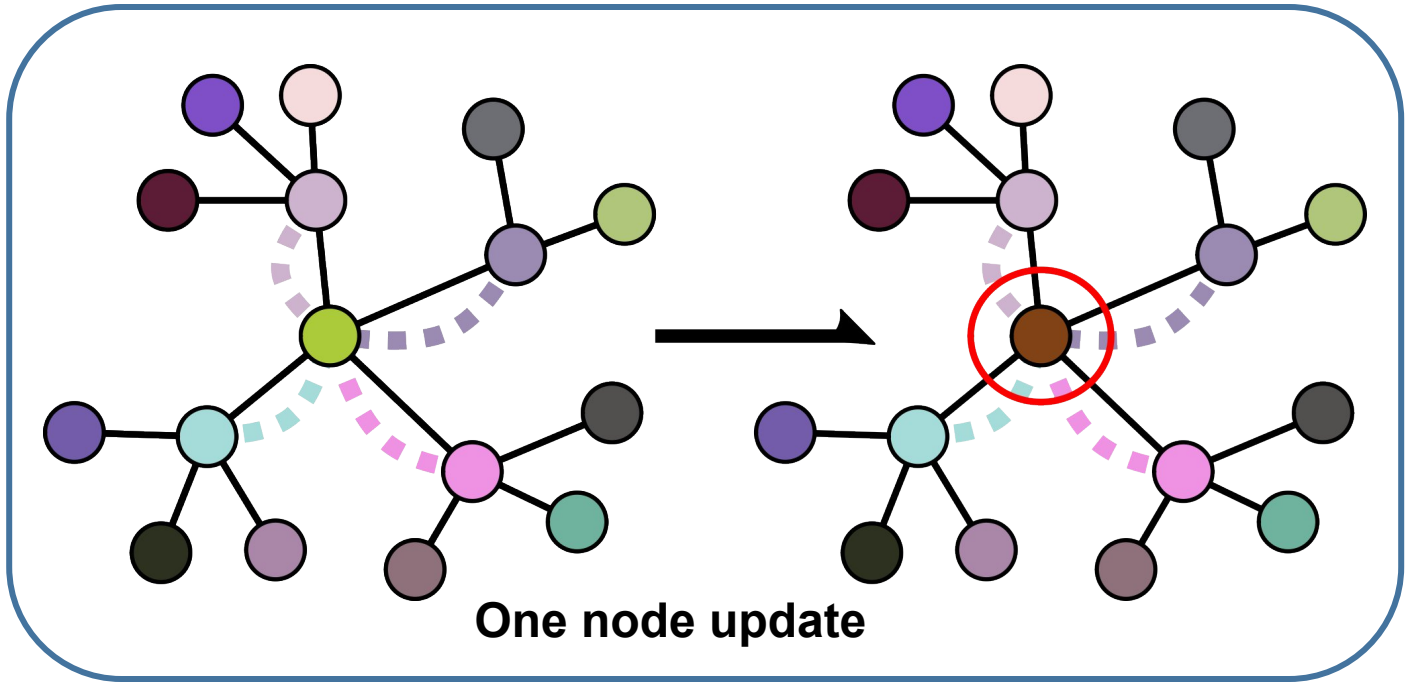




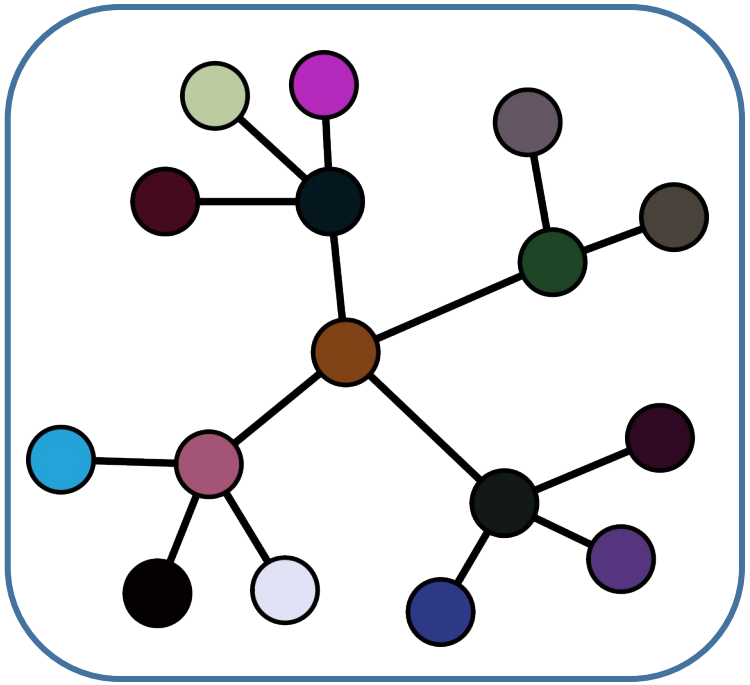
Chami, I.; Abu-El-Haija, S.; Perozzi, B.; Ré, C.; Murphy, K. Machine Learning on Graphs: A Model and Comprehensive Taxonomy. *J. Mach. Learn. Res.* **2022**, *23* (1), 89:3840-89:3903.

Graph Convolution

Just like for images we can learn from neighbourhood with a convolution



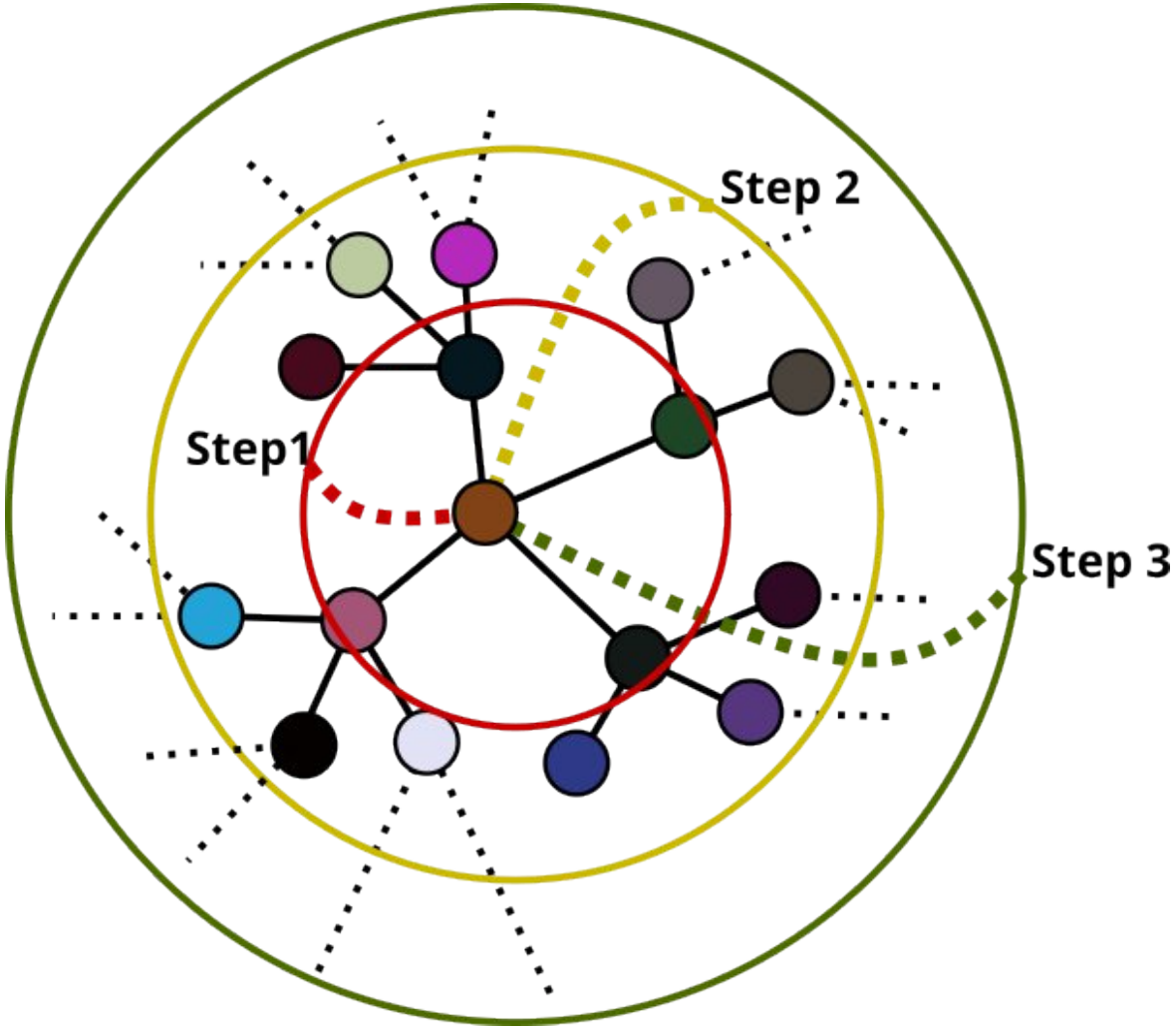
Full convolution:
All nodes representation updated



- A bit more complex since the number of neighbours is unlikely to be constant
- We want the operator to be permutation invariant
- One step = information from direct neighbours
- We use adjacency matrix

Graph Convolution

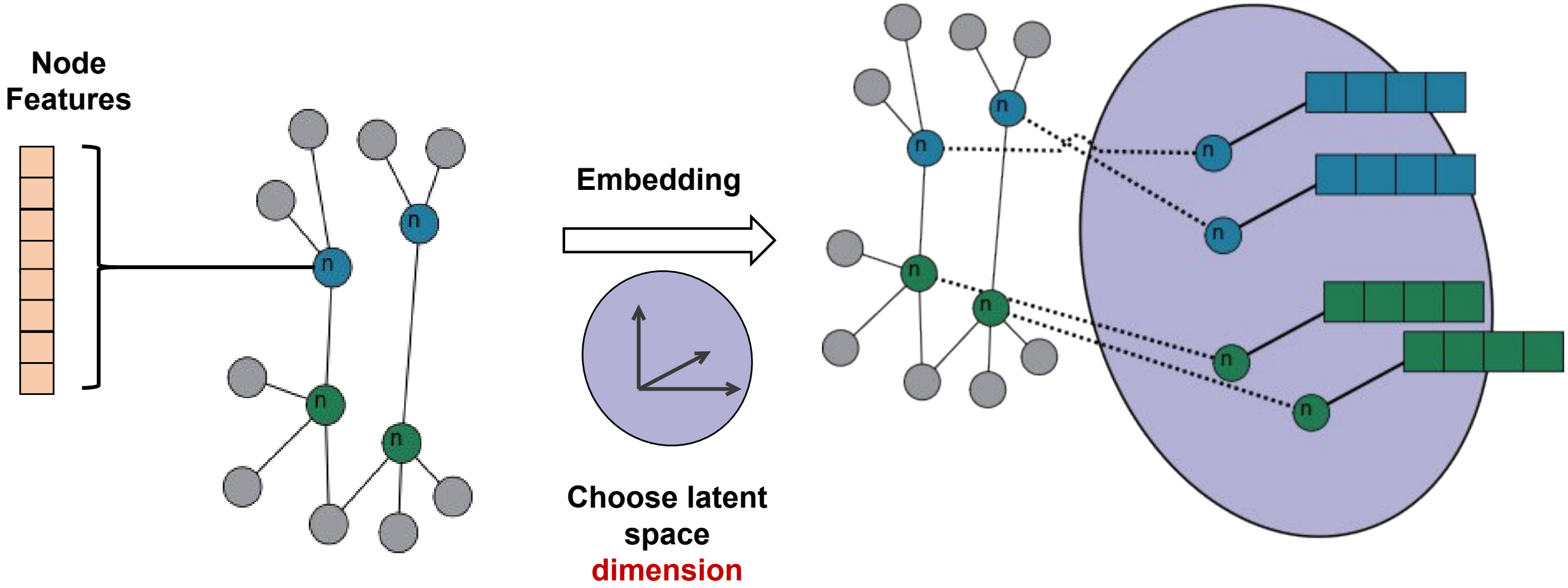
Get information from further neighbours: Add Conv. Layers



- Several **layers** are needed to retrieve information for distant nodes
- Each layer brings information from nodes further on a path
- For large graphs → a **cutoff**
- It is possible to use a **virtual node** connected to all other nodes. But in practice this becomes quickly intractable.

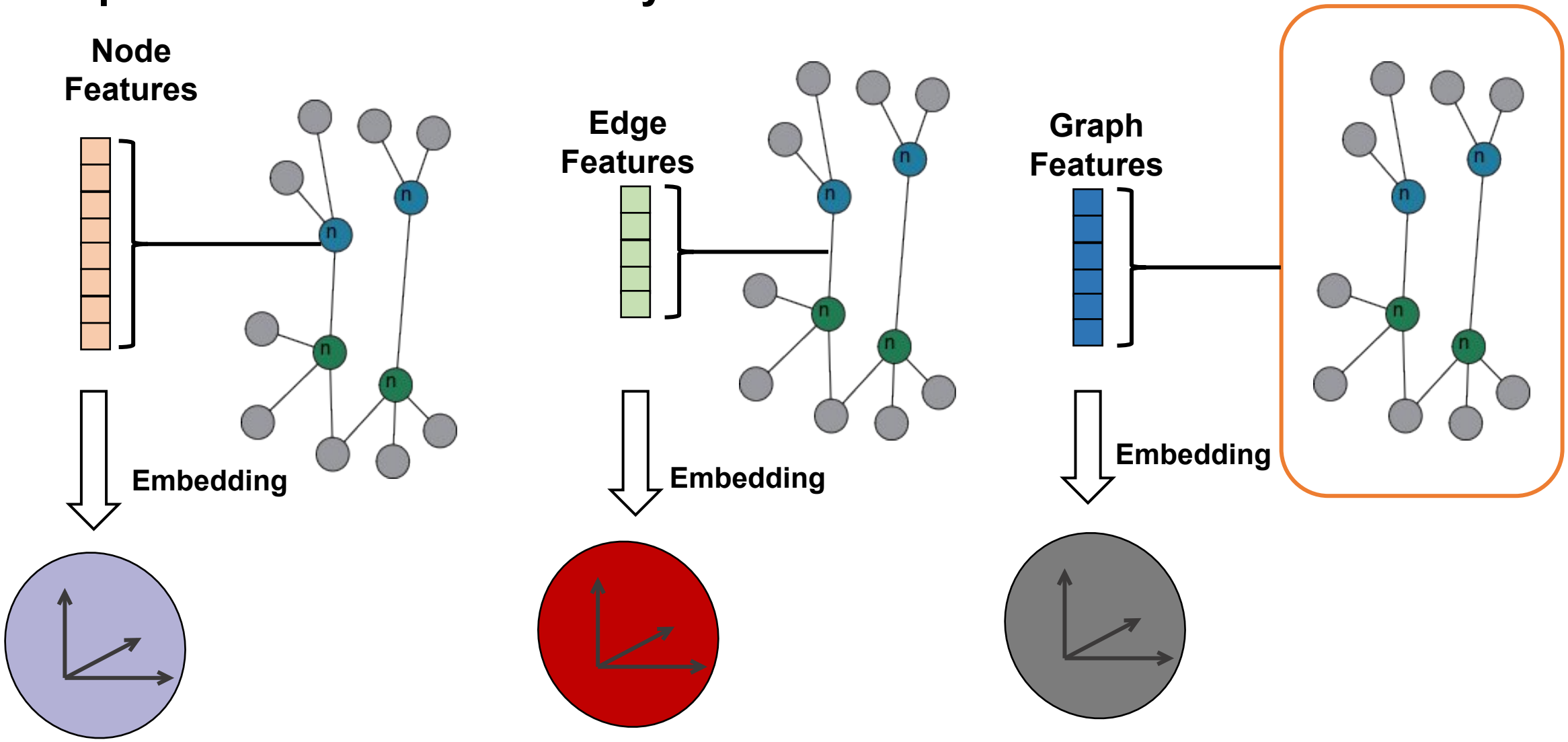
Node Embedding: Representing features (reminder)

Each node is represented in the latent space



Edge and Graph embedding

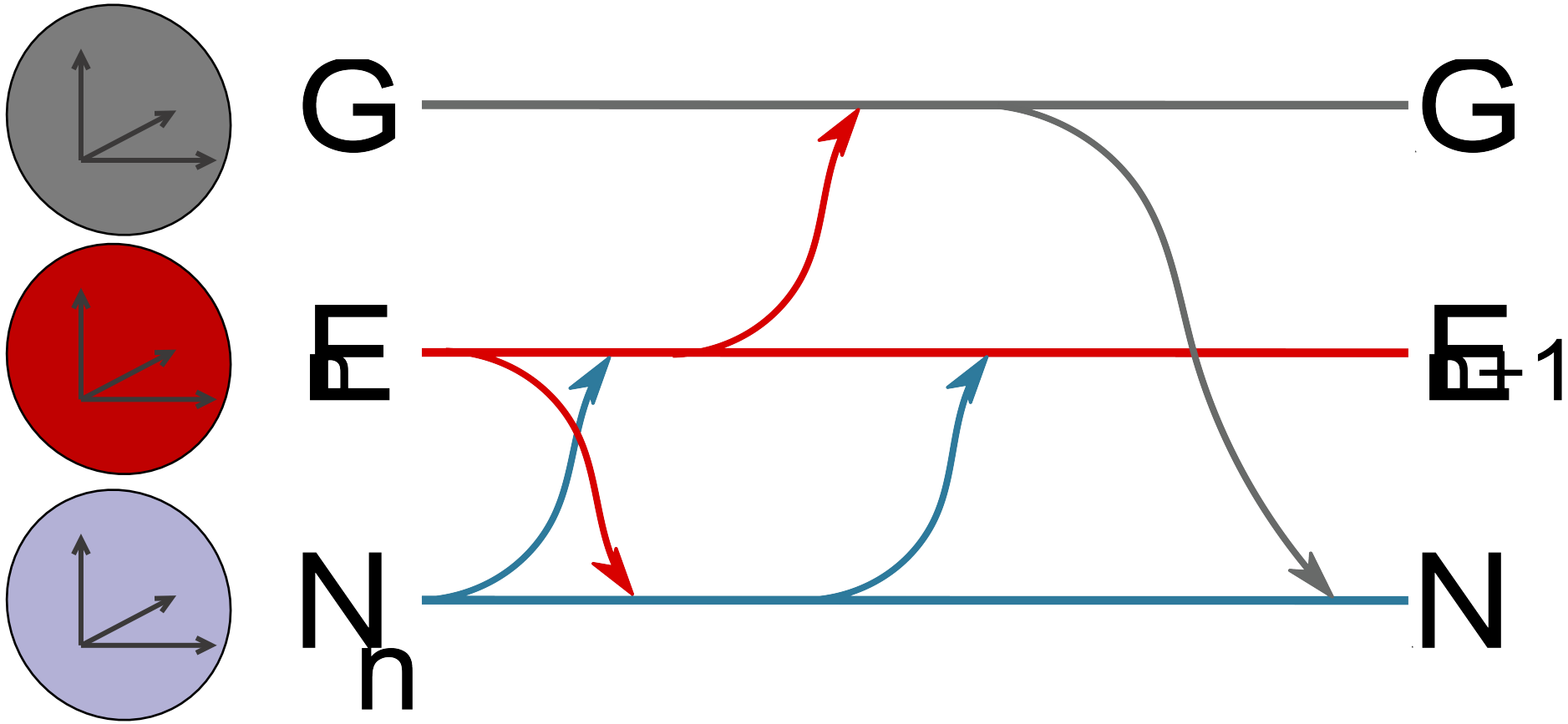
Graphs store information everywhere !



Even more hyperparameters!

Message passing

We can share information between all parts of the graph

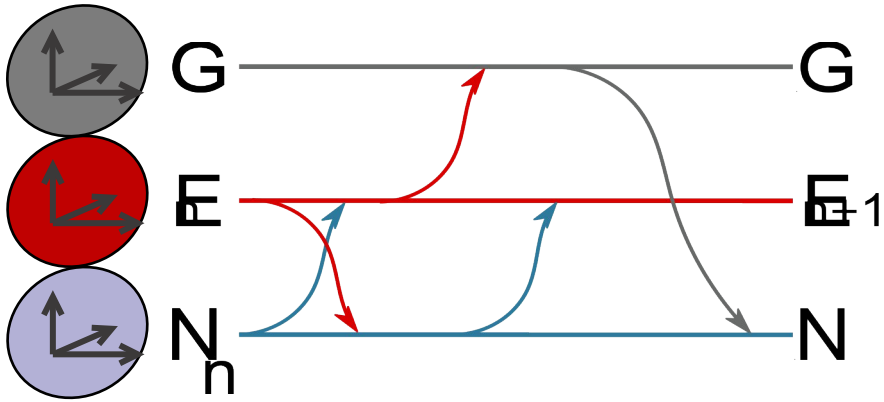


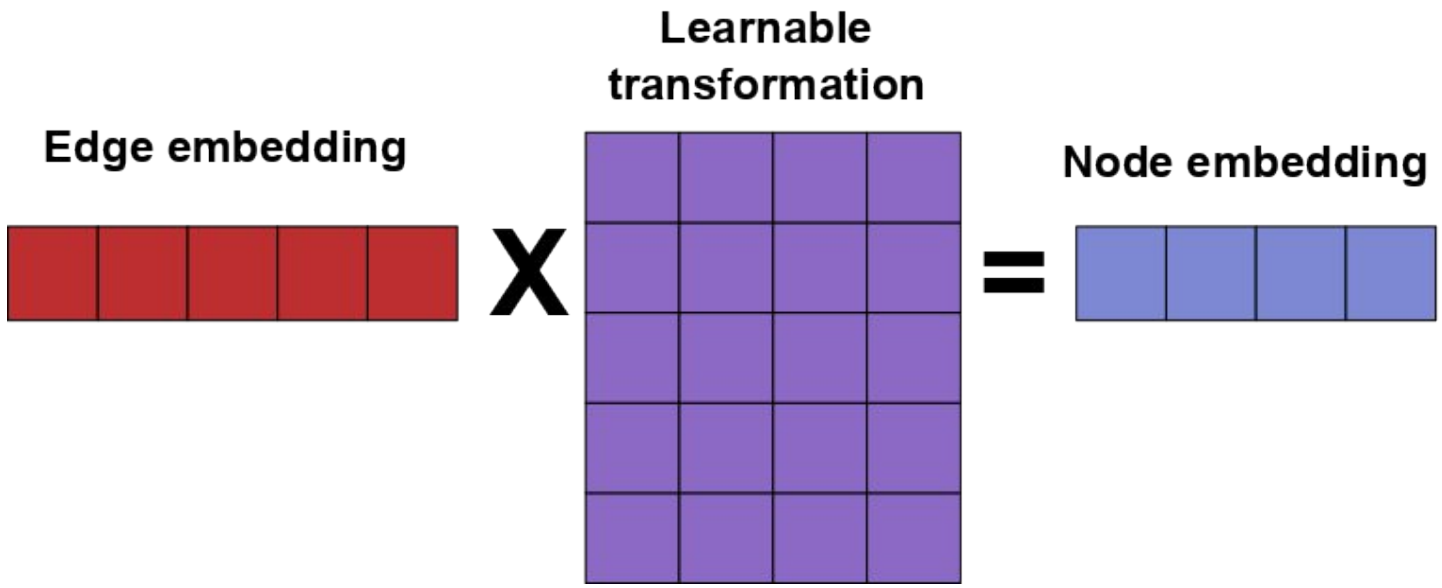
Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; Dahl, G. E. Neural Message Passing for Quantum Chemistry. arXiv June 12, 2017. .

Message passing: transform

We can share information between all parts of the graph

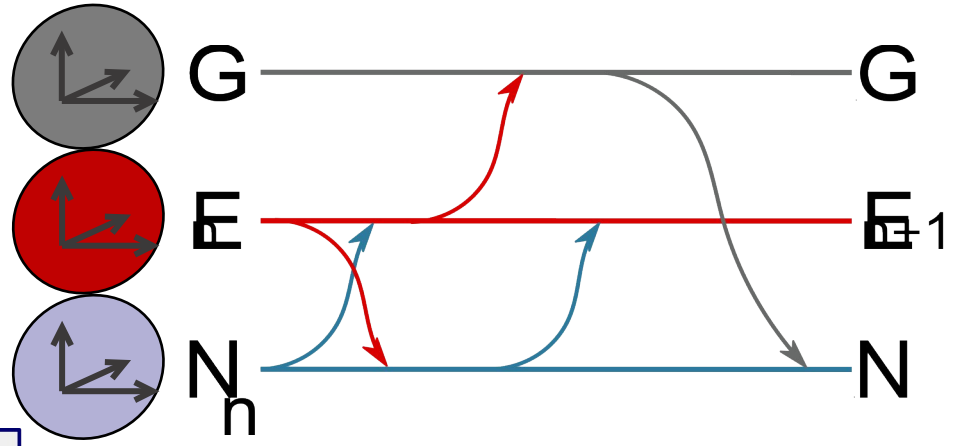
- We need a **transformation** to exchange since vectors are not of the same size
- The transformation is **learnable**



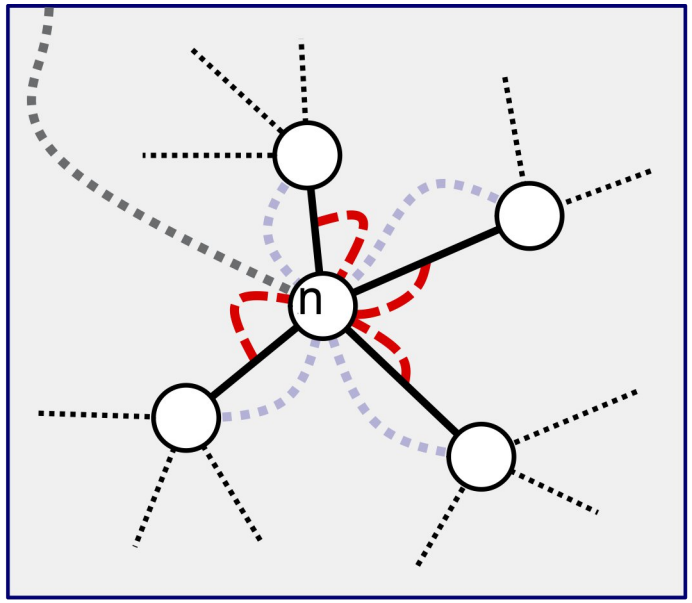


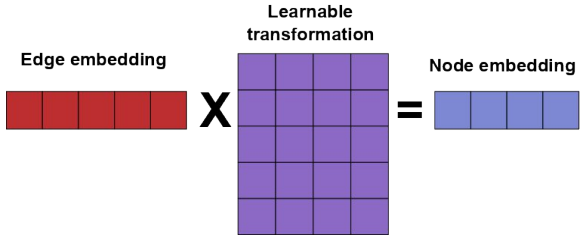
Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; Dahl, G. E. Neural Message Passing for Quantum Chemistry. arXiv June 12, 2017. .

Message passing: build the message



- We build the message by **aggregation** of the latent representations
- The aggregation has to be **invariant** by **permutation** of the nodes



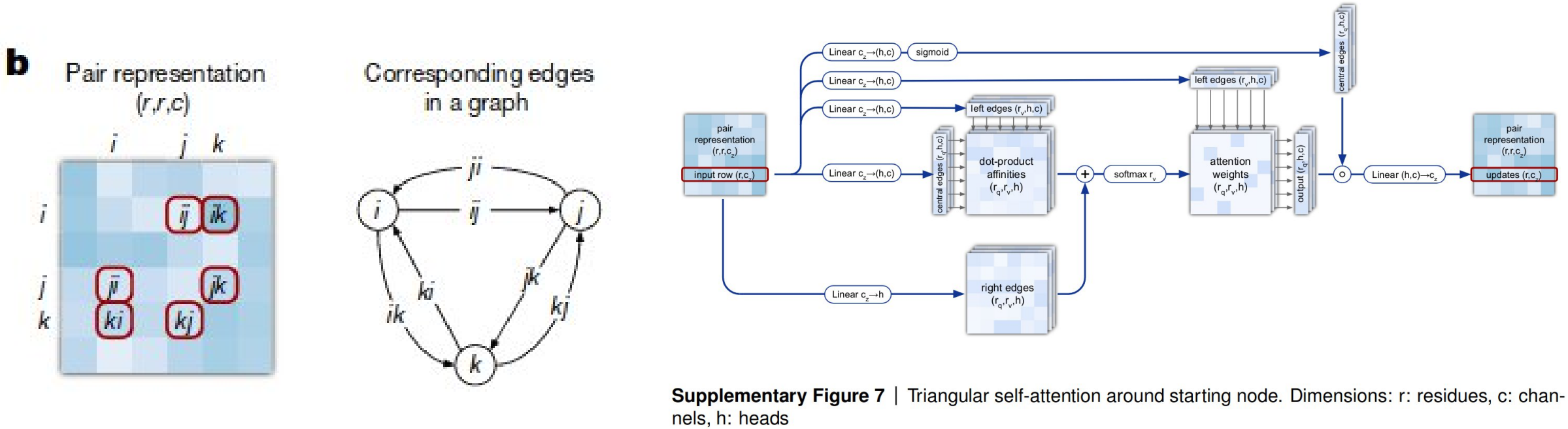


$$a' = \left(\text{aggr} \quad f_N \right) , f_E , f_G)$$

Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; Dahl, G. E. Neural Message Passing for Quantum Chemistry. arXiv June 12, 2017. .

Alphafold transformer

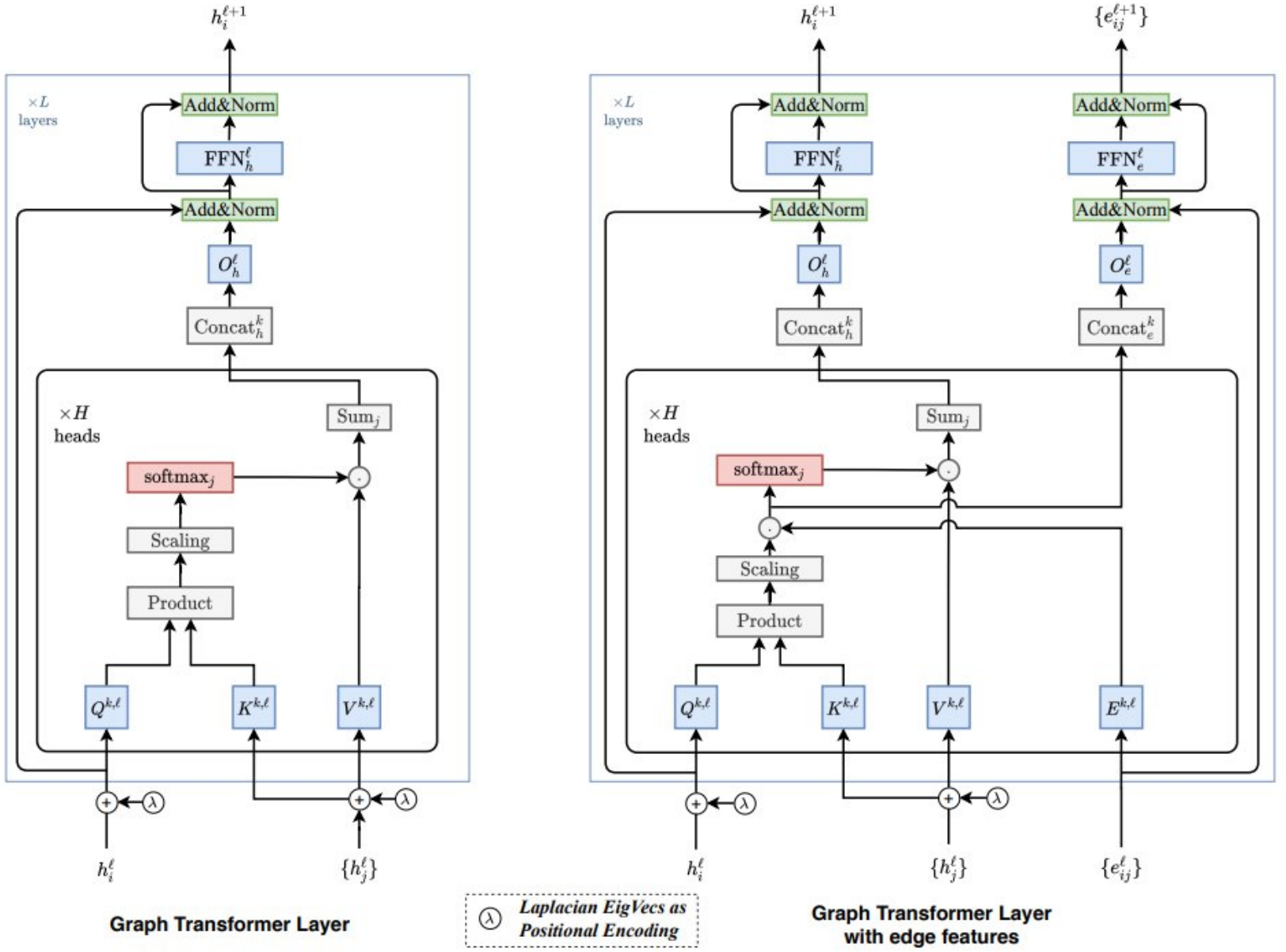
Alphafold uses a kind of adjacency matrix in the evoformer block



Supplementary Figure 7 | Triangular self-attention around starting node. Dimensions: r: residues, c: channels, h: heads

Jumper, J.; Evans, R.; Pritzel, A.; Green, T.; Figurnov, M.; Ronneberger, O.; Tunyasuvunakool, K.; Bates, R.; Žídek, A.; Potapenko, A.; Bridgland, A.; Meyer, C.; Kohl, S. A. A.; Ballard, A. J.; Cowie, A.; Romera-Paredes, B.; Nikolov, S.; Jain, R.; Adler, J.; Back, T.; Petersen, S.; Reiman, D.; Clancy, E.; Zielinski, M.; Steinegger, M.; Pacholska, M.; Berghammer, T.; Bodenstein, S.; Silver, D.; Vinyals, O.; Senior, A. W.; Kavukcuoglu, K.; Kohli, P.; Hassabis, D. Highly Accurate Protein Structure Prediction with AlphaFold. *Nature* **2021**, *596* (7873), 583–589. .

Graph Transformer Network



12

«Attention is All You Need»
RNN, Transformers

Dwivedi, V. P.; Bresson, X. A Generalization of Transformer Networks to Graphs. arXiv January 24, 2021. .