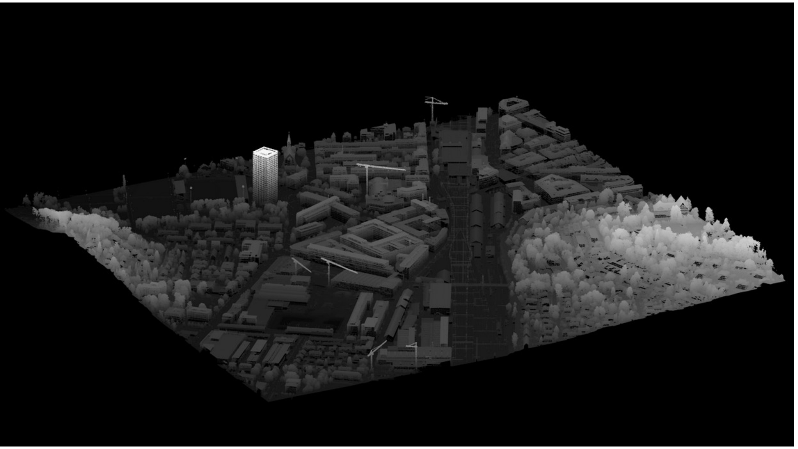
Graphs are everywhere

Highly ordered data

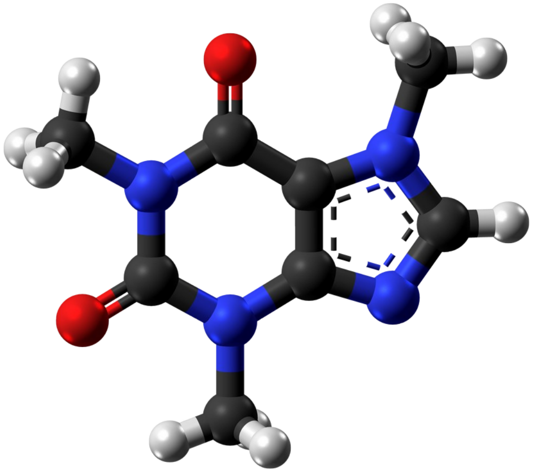
Rebirth of Deep learning was thanks to pictures, text and speech recognition

Data structures: Data is not always euclidean

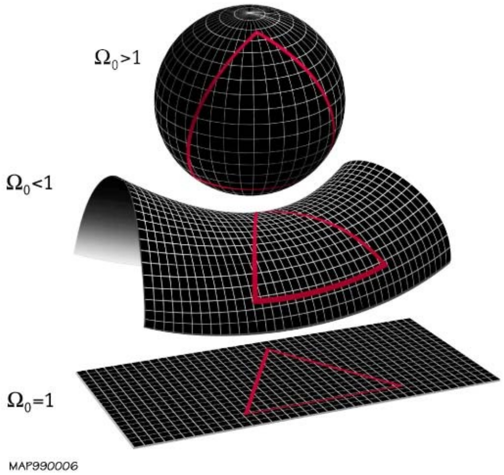
LIDAR



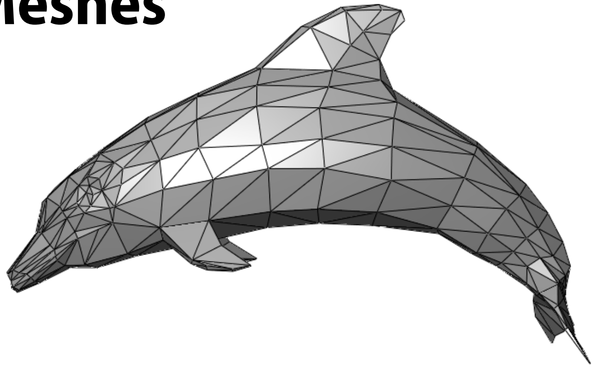
Molecules



Complex geometries



Meshes

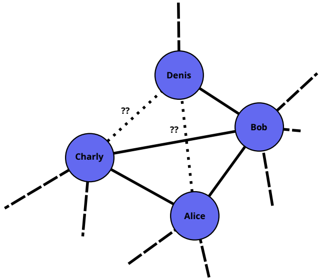


Geometric deep learning

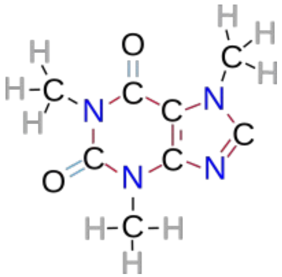
Michael M. Bronstein, Joan Bruna, Taco Cohen, Petar Veličković, Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges https://arxiv.org/abs/2104.13478

Graphs are everywhere

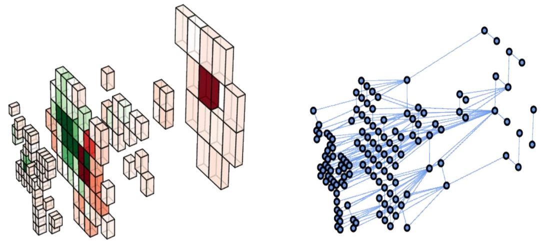
Social networks



Molecules



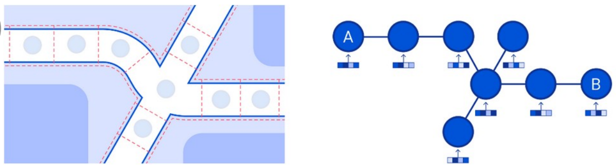
Particle physics



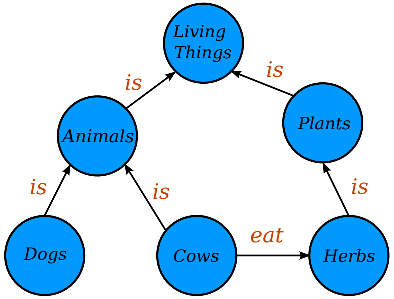
[2]

Directions recommendation

[1]



Knowledge graphs

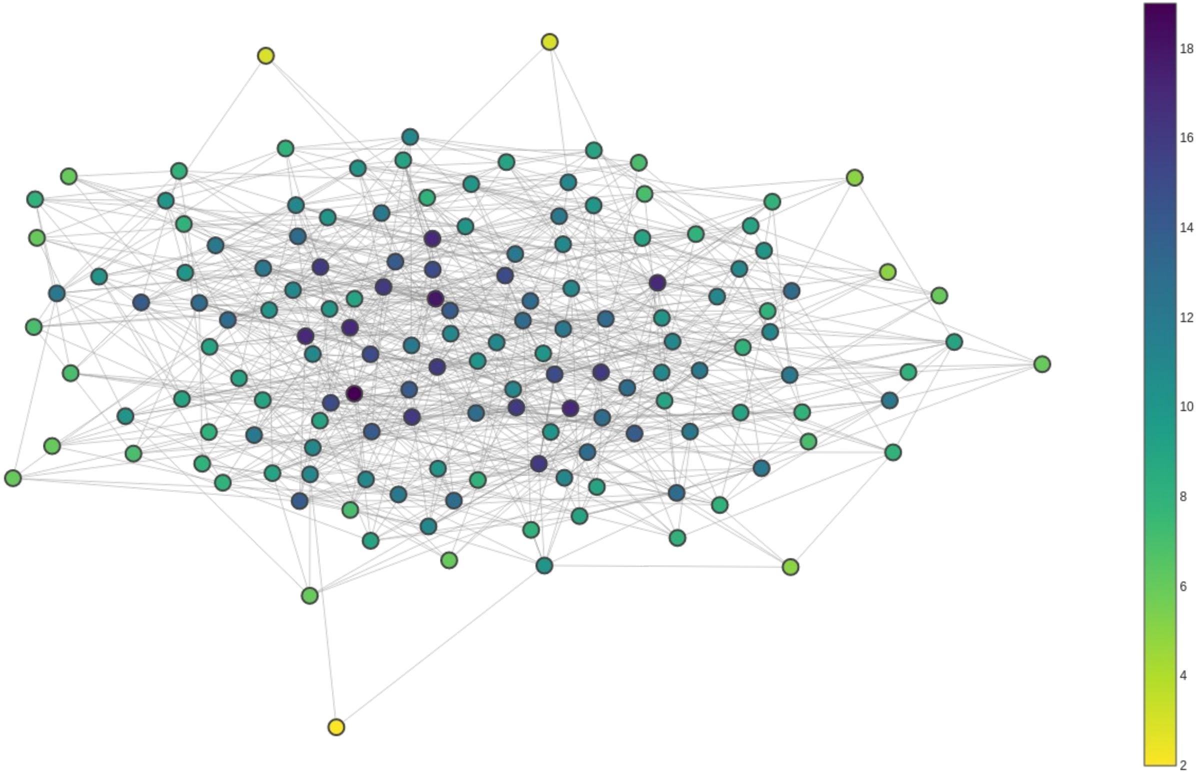


Many other fields

- Biology
- Recommendation systems
- Computer vision
- Medical diagnosis
- Robotics
- ...

[1] A. 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.
[2] J. Shlomi, P. Battaglia, and J.-R. Vlimant, "Graph neural networks in particle physics," Mach. Learn.: Sci. Technol., vol. 2, no. 2, p. 021001, Jan. 2021, doi: 10.1088/2632-2153/abbf9a.

Complexity

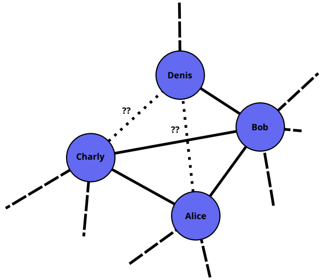


- Might have several thousand nodes/edges
- Number of edges/nodes might vary a lot
- ...

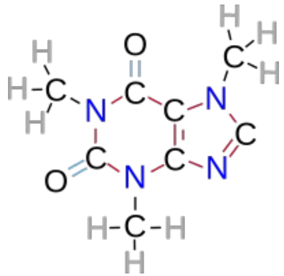
Vocabulary: Node/Vertex

Some example of nodes

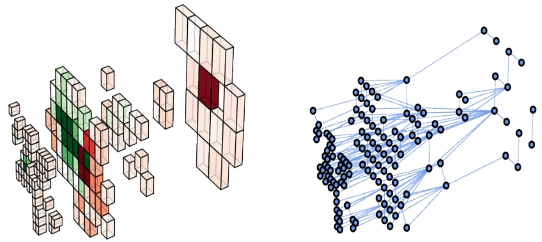
Persons



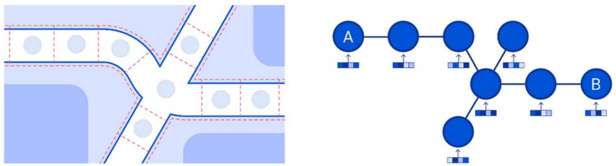
Atoms



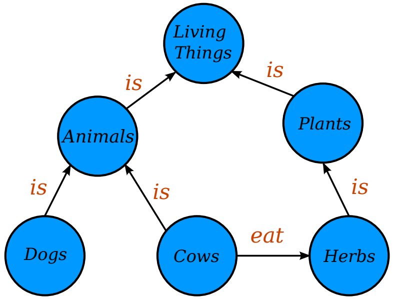
Particles



Road sections



A concept



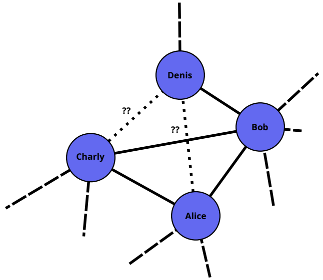
Many other fields

- Biology : an aminoacid in a protein
- Recommendation systems : a customer
- Computer vision : an element in a picture
- Medical diagnosis : Brain region (MRI)
- Robotics : joints
- ...

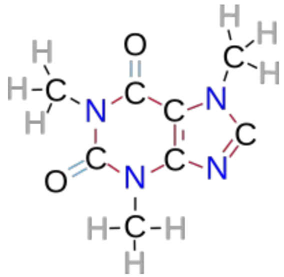
Vacabulary: Edges

Some example of nodes

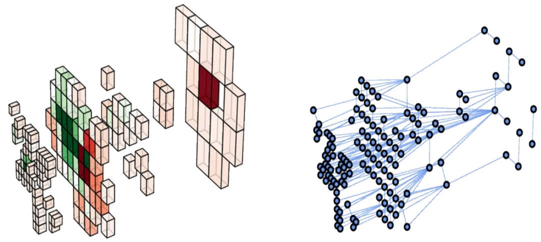
Relationship



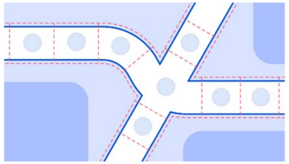
Type of bond

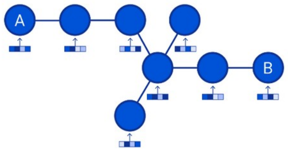


Decayed to

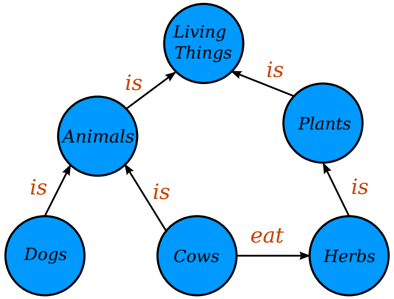


Time, connection





A statement



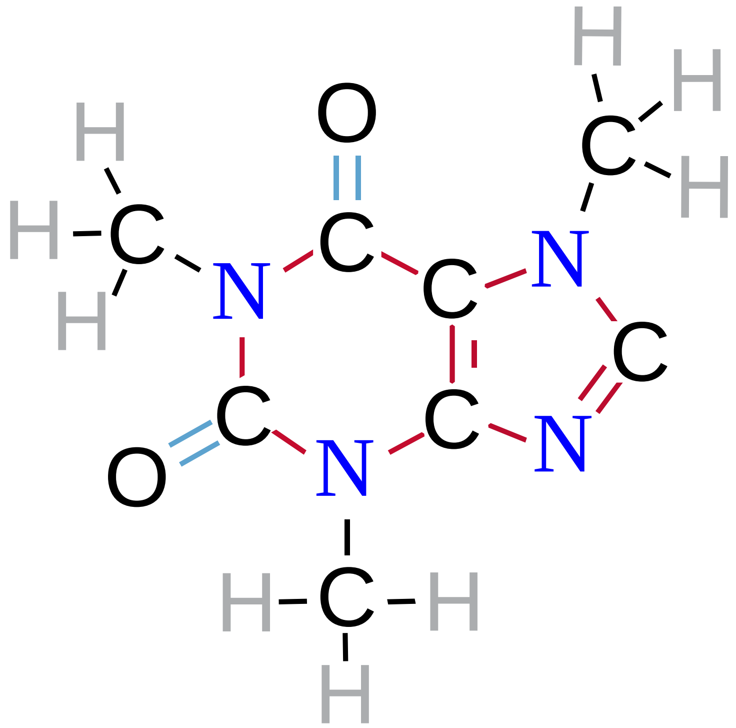
Many other fields

- Biology : distance between residues
- Recommendation systems : connected customers
- Computer vision : an interaction between elements
- Medical diagnosis : interaction between brain regions (MRI)
- Robotics : connection between joints
- ...

Vocabulary: Edges orientation

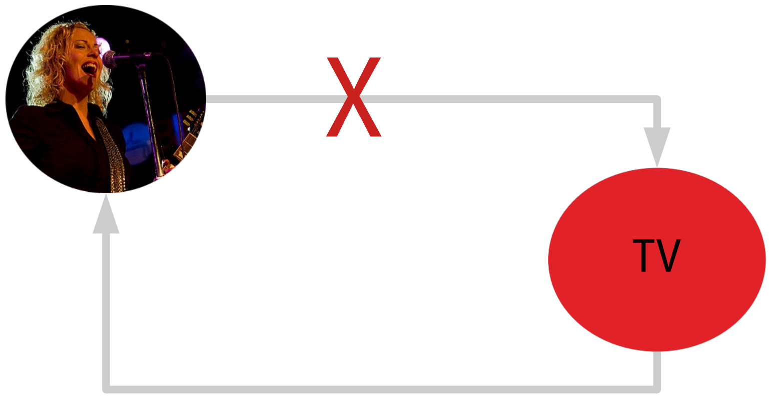
A relationship can be symmetrical or not between nodes

Undirected graphs



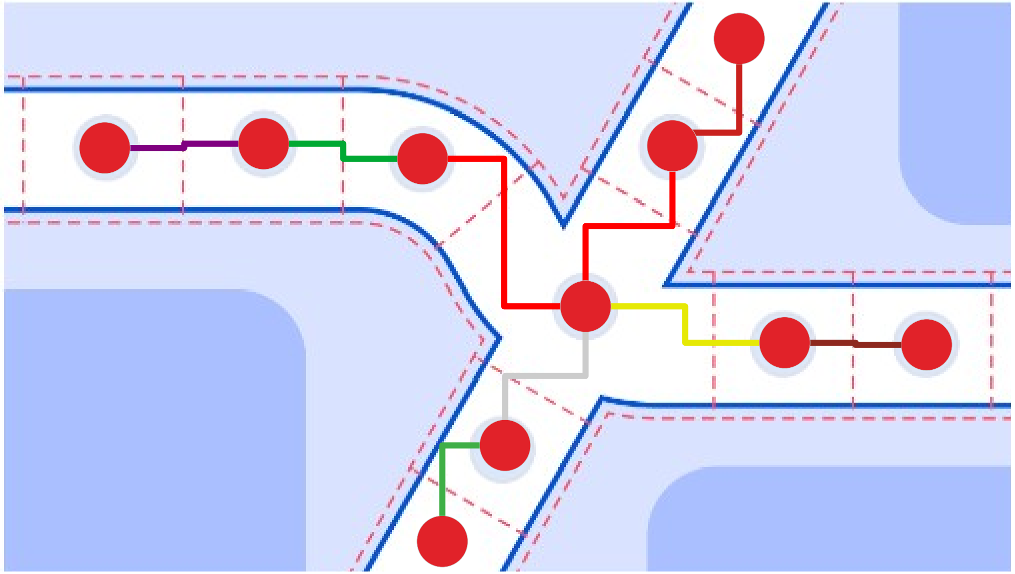
Directed graphs

Anneke van Giersbergen



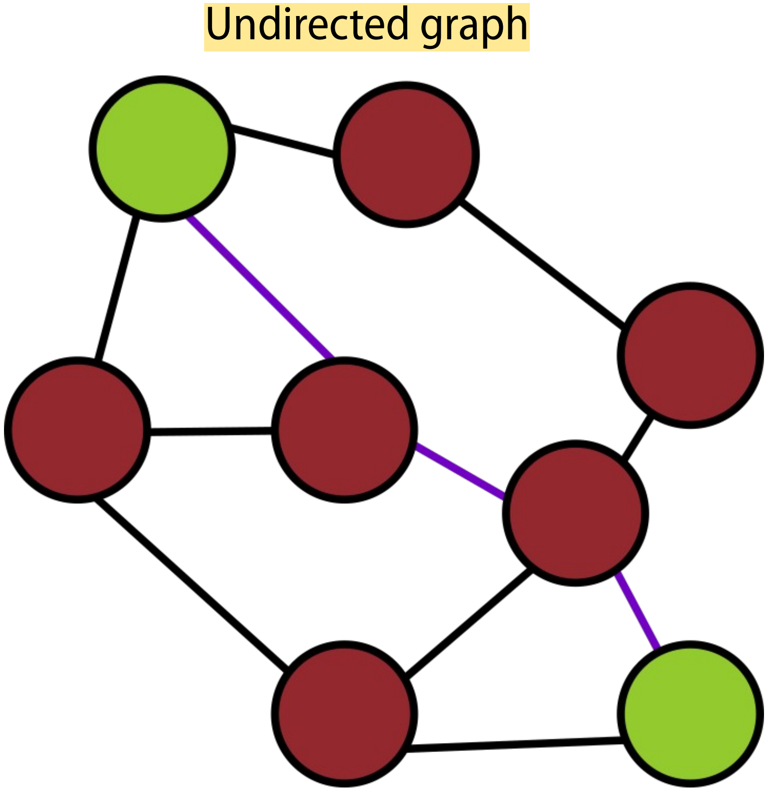
Vocabulary: Edges weight

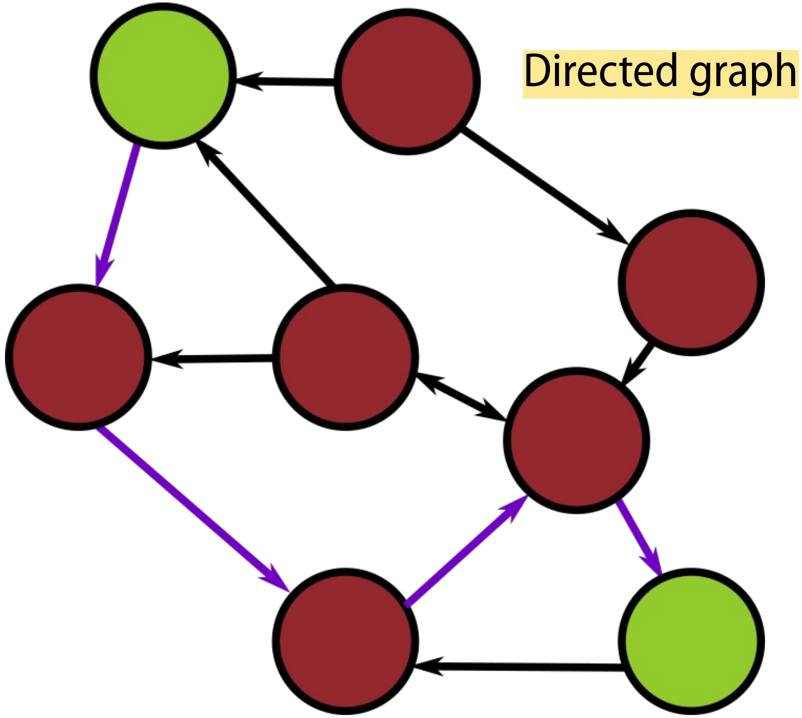
Edges can carry more information



Vocabulary: Paths

A path is a sequence of edges connecting 2 nodes





Formal definition

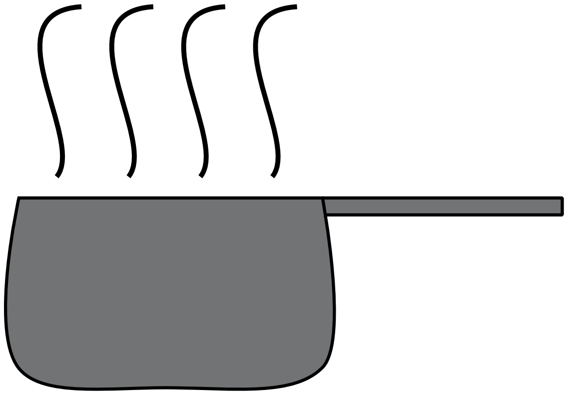
G = (V, E): a set of nodes and edges

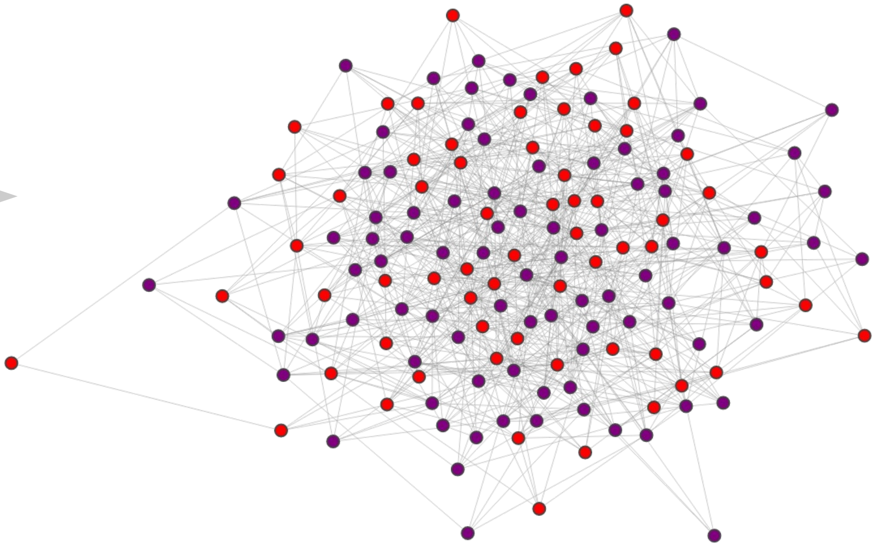
Labels

$\{y_i^N\}\ \{y_i^E\}\ \{y_i^G\}$

$\{v_i\}_{i \in V}$

$\{e_i\}_{i \in E}$





Graphs store information: Labels

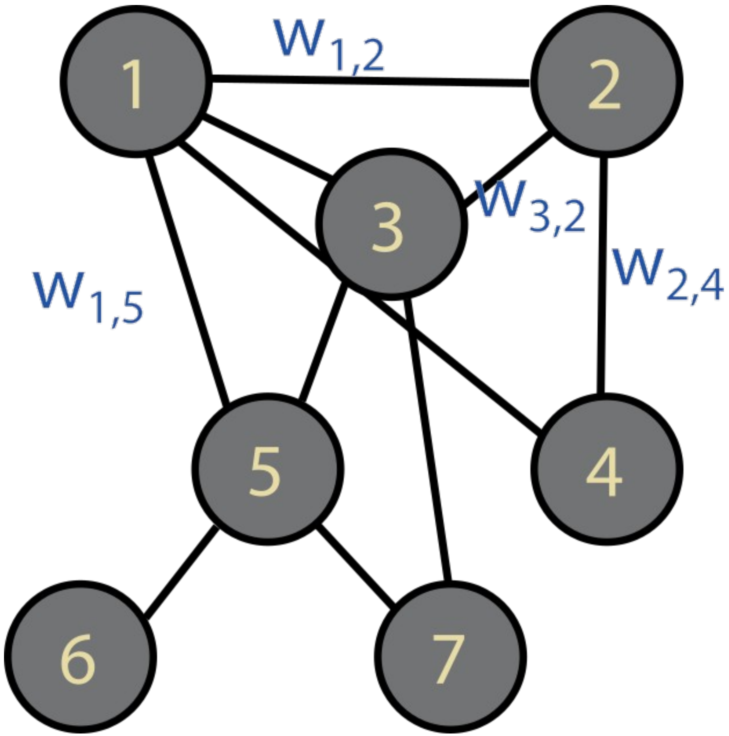
- Graphs can store information (features) on **nodes**, **edges** and **globally**

	Globally	Nodes	Edges
Social Network	Group of interest, ...	Name, Age, Job, ...	Is friend, follows, family, ...
Molecule	Is a drug, Energy, ...	Atomic number, ...	Bond order, ...
Citations	Field, ...	Article, ...	Was cited, ...
Particle physics	Experiment	Particle	Decayed to, ...
Motion capture	Character	Joints	Is connected to, ...
Natural language	Paragraph, ...	Group of words, ...	Refers to, ...

- It can be a number, a concept, ...

Node proximity and centrality

Measure of the structure of a graph



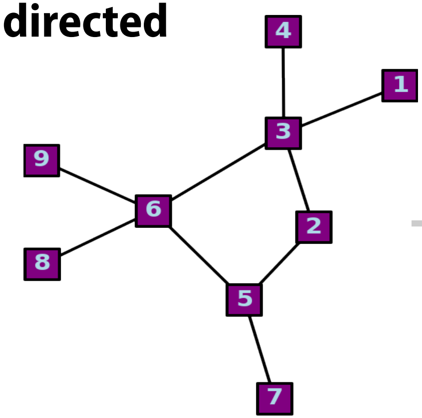
Node proximity
- 1st order: $\mathbf{w}_{i,j}$ between node i and j
- 2nd order: similarity of neighborhood structure
- Higher orders possible

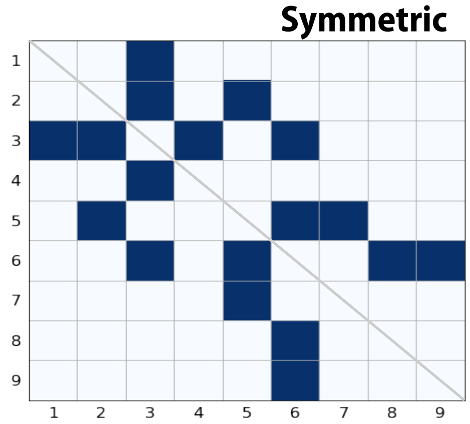
Node centrality
- Measure how many paths goes through the node

Graph representation

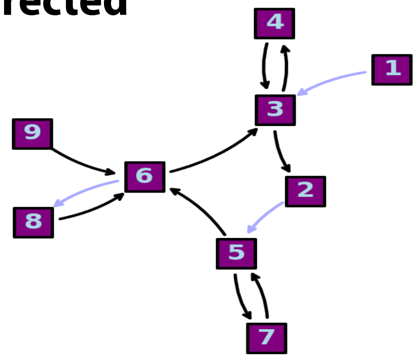
Adjacency matrix $W_{(i,j)} = \begin{cases} w_{i,j} \text{ if there is an edge} \\ 0 \text{ if not} \end{cases}$

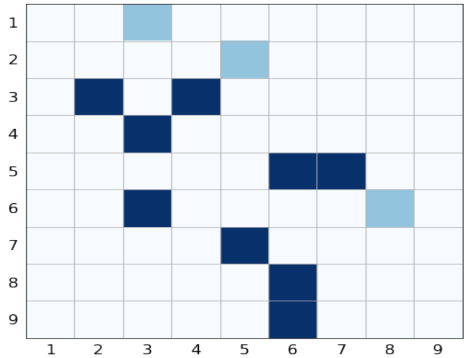
Undirected



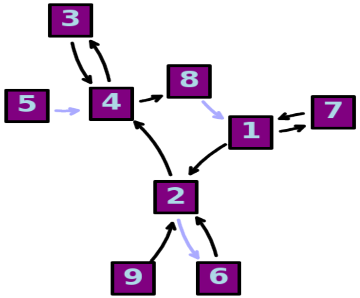


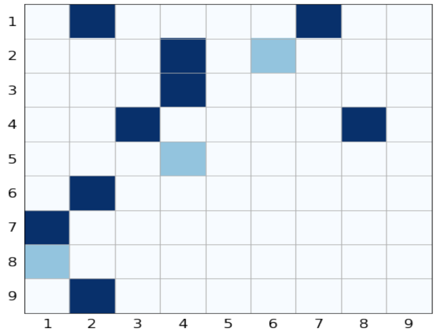
Directed





Adjacency list





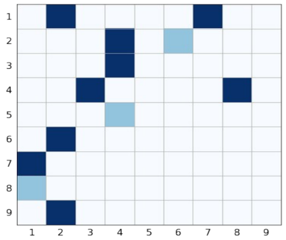
Nodes: [1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0]

Edges: [0.4, 0.4, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 0.4, 1.0, 1.0, 1.0]

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

Global: [1.0, 1.0]

Graph representation



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

- Scale V**2 → lot of space
- Sparse
- N ! permutations to represent the same graph

- Easy to find an edge

- Scale E → less space

- Might be difficult to find an edge

V = number of nodes/vertices
E = number of edges

https://www.geeksforgeeks.org/comparison-between-adjacency-list-and-adjacency-matrix-representation-of-graph/

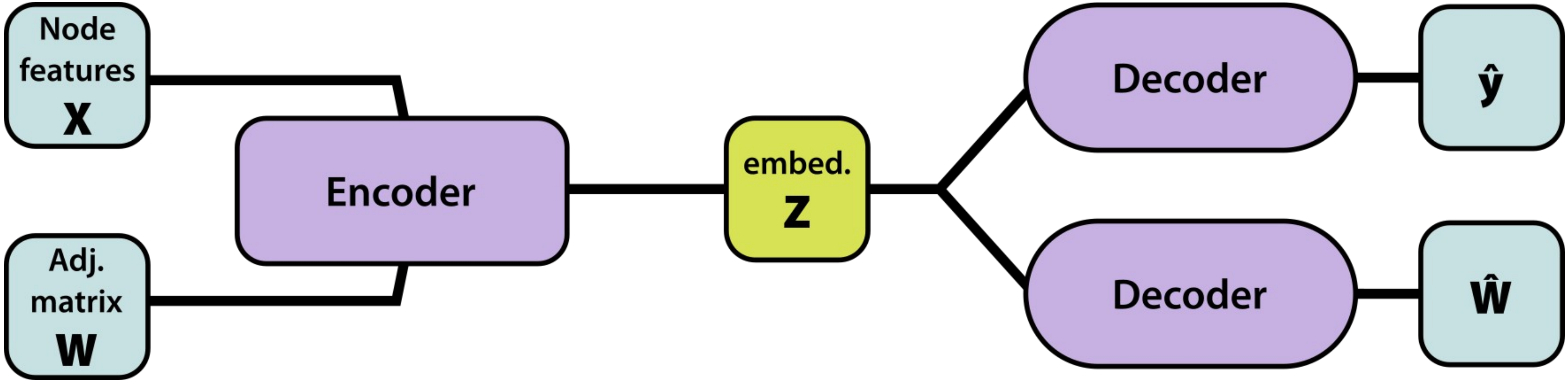
Useful Matrices

Adjacency	**W**	Weight of edges
Degree	**D**	Diagonal matrix with number of edges for each node
Laplacian	**L**	**D** - **W**
Node Features	**X**	Information stored

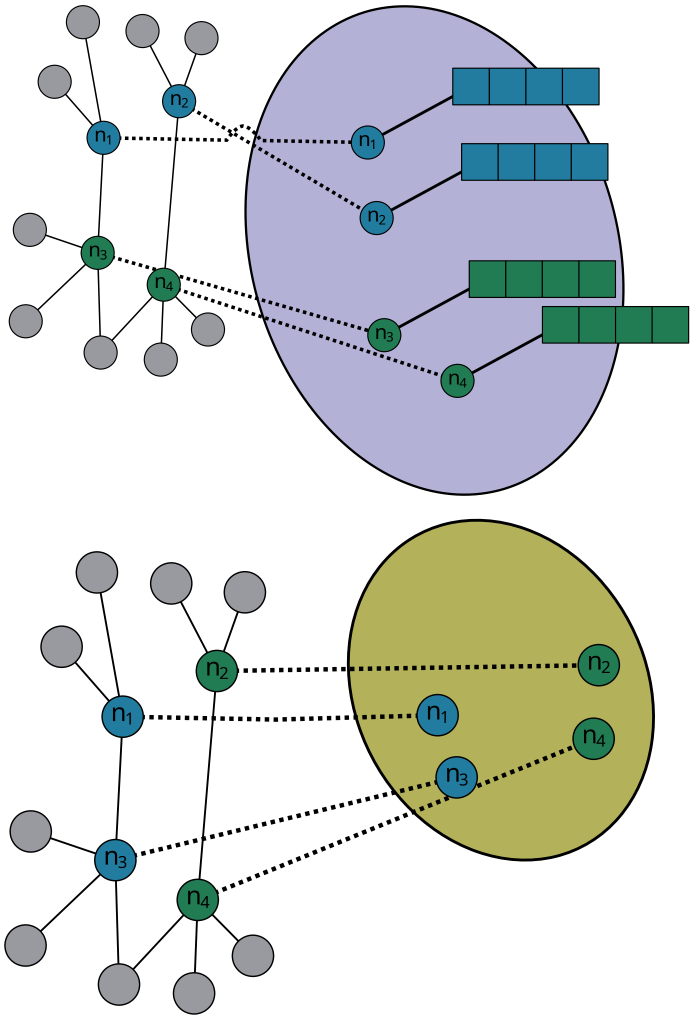
Learning on Graphs

Graph embedding

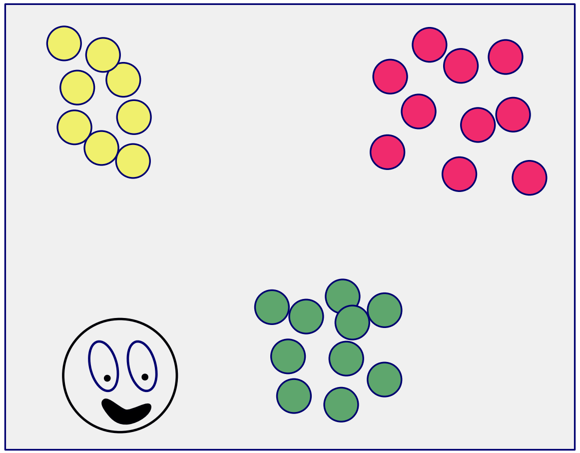
- We need to find a representation of the graph that is processable

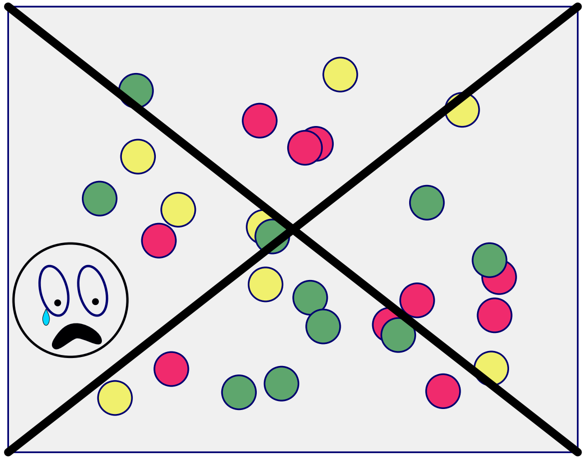


Graph embedding



- 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.**



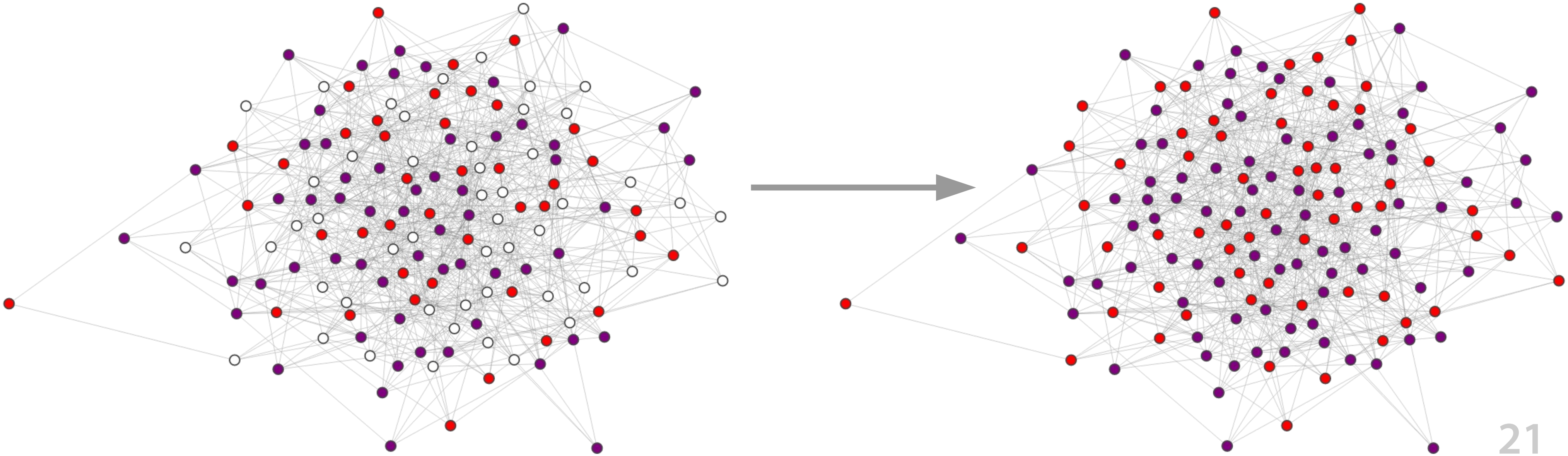


Transductive learning

The model has access to the complete graph

It is not possible to add new nodes

Node labeling

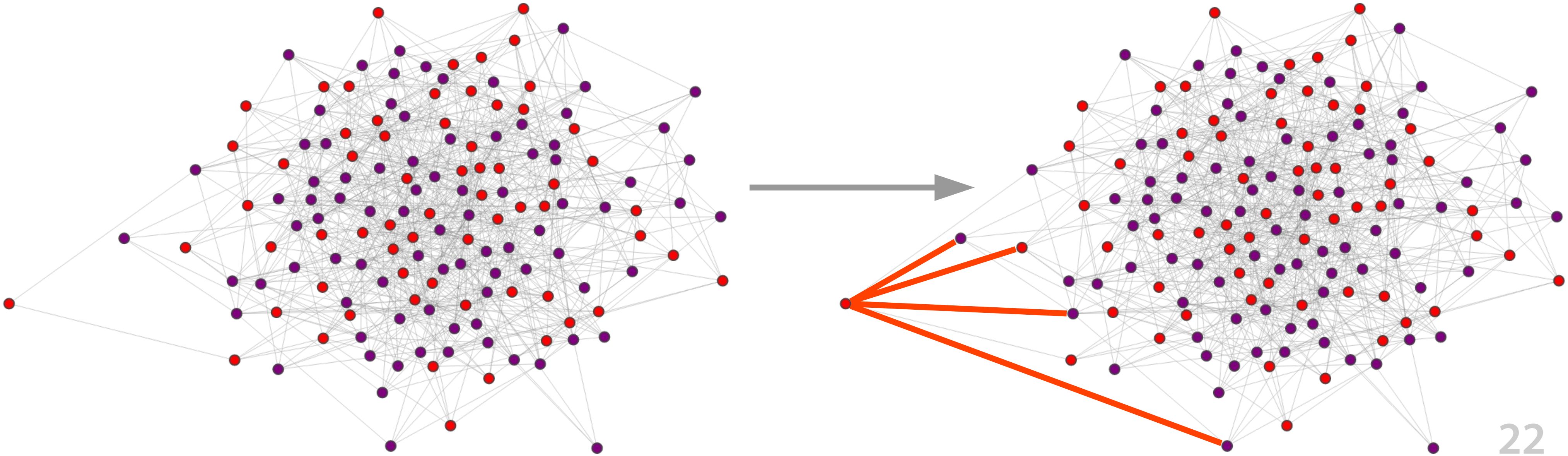


Transductive learning

The model has access to the complete graph

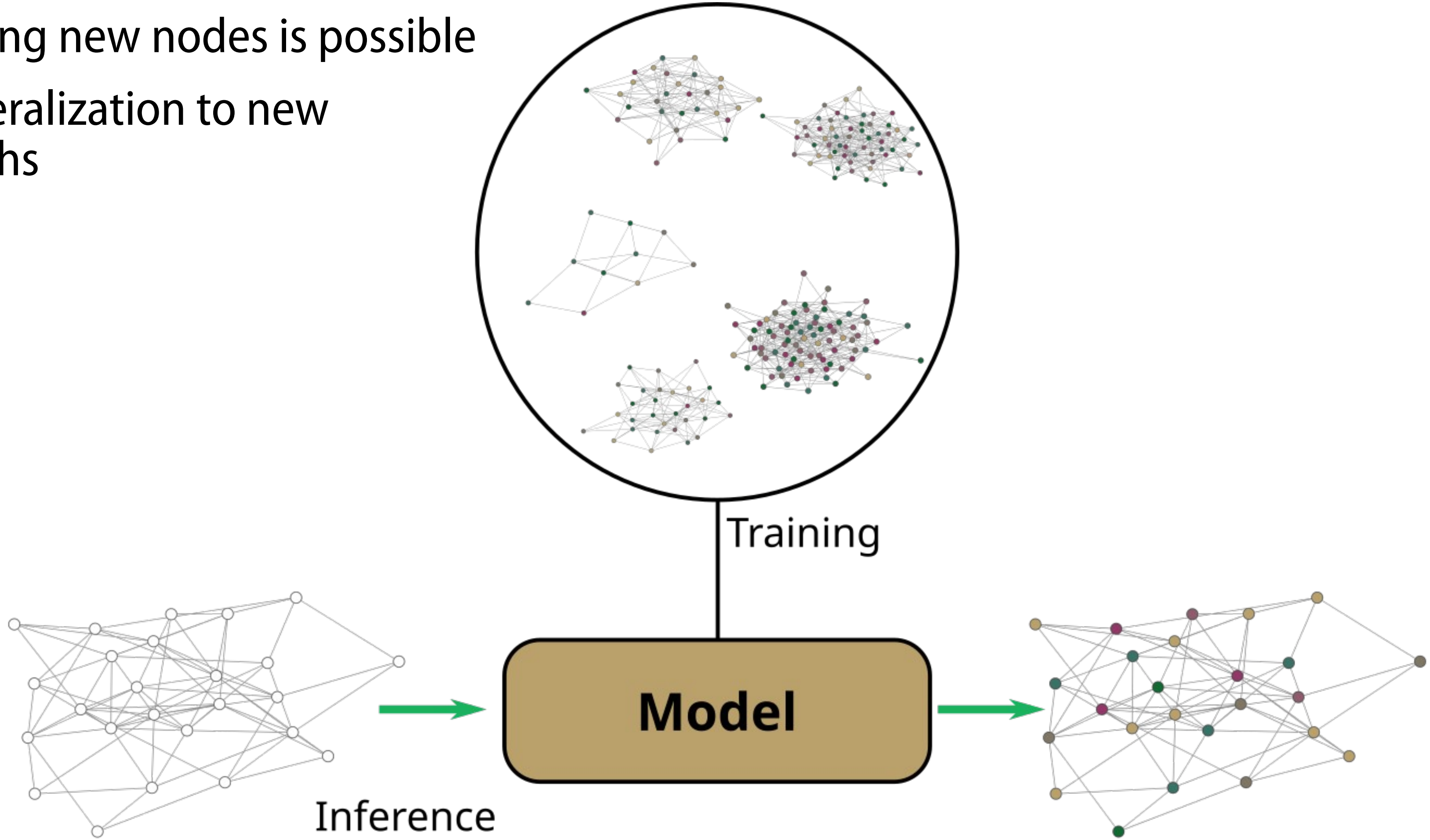
It is not possible to add new nodes

Find new edges



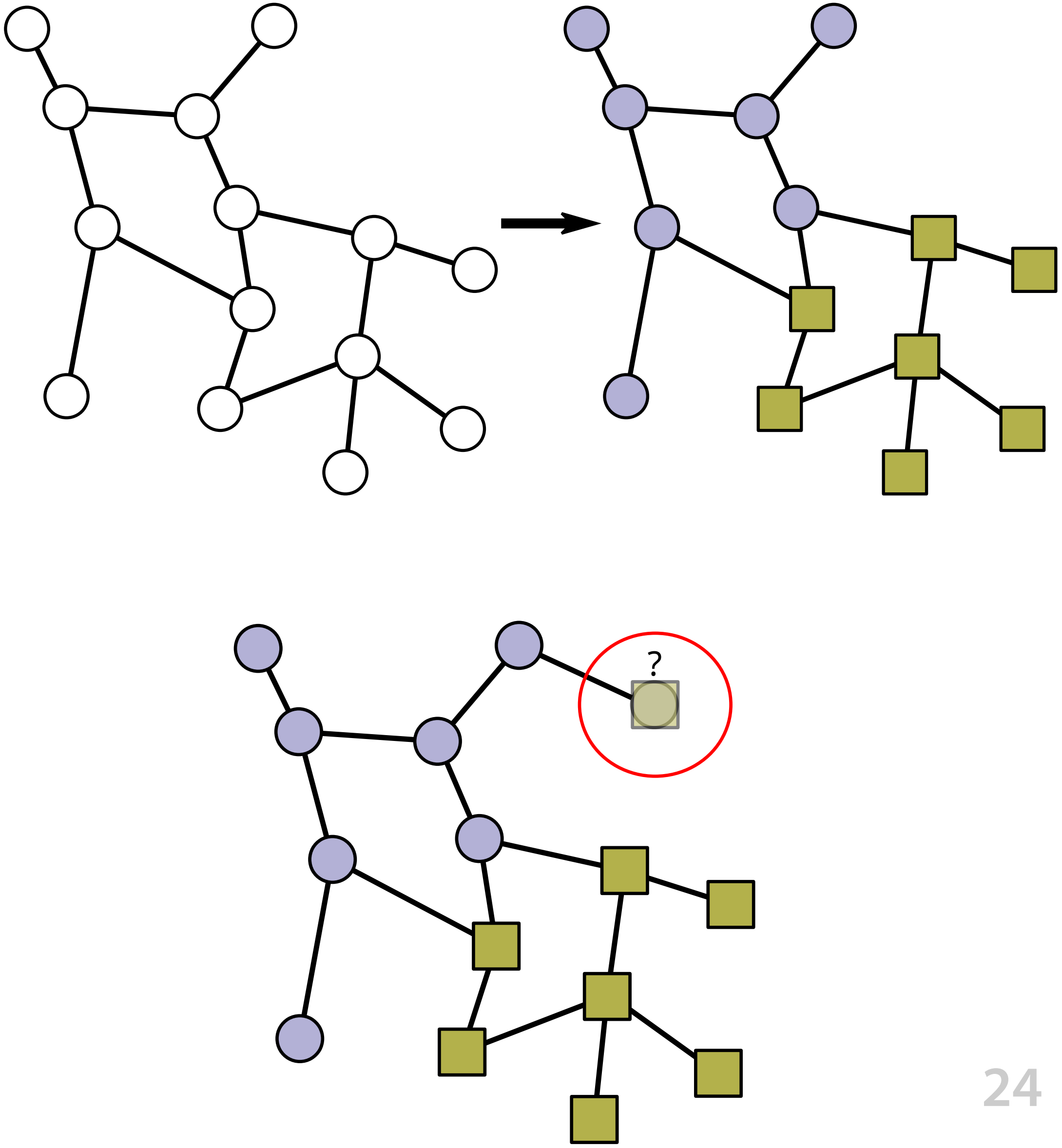
Inductive learning

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



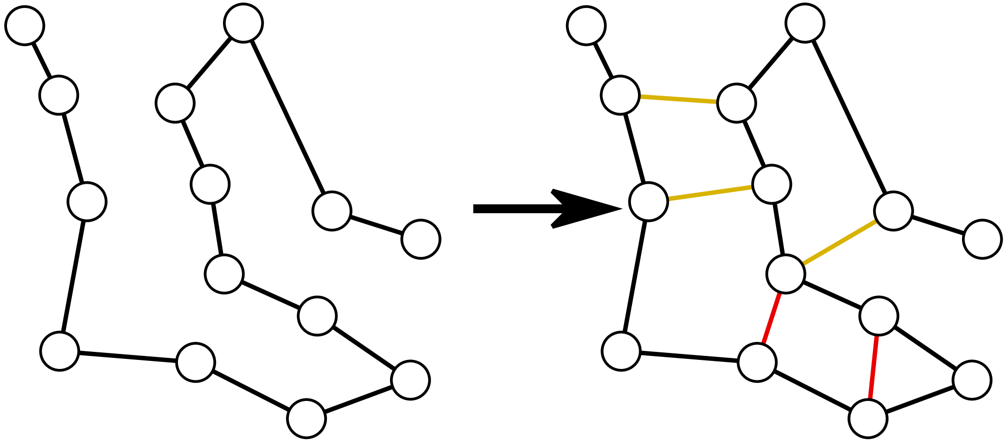
Tasks on nodes

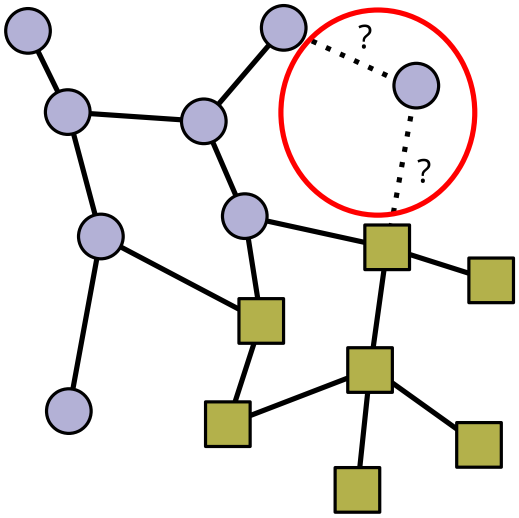
- Labeling nodes in a graph (clustering)
  - Find topic of a research paper (CORA, etc)
  - Find bots in a social network
  - …
- Labeling new nodes
- Perform regression

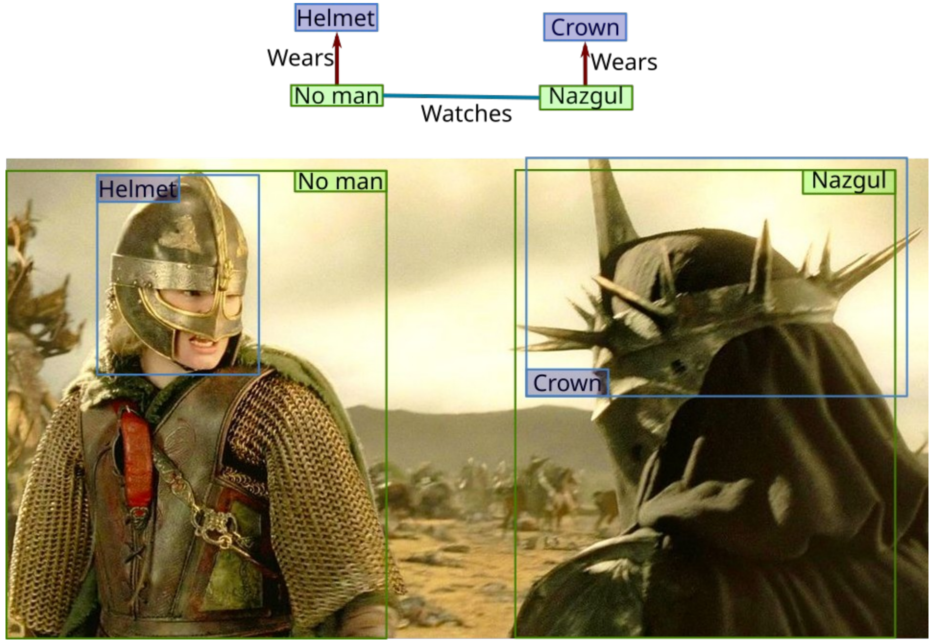


Tasks on edges

- Find relationships
  - Contact map of aminoacids (Alphafold)
  - Contact suggestion (social network)
  - ETA for directions (regression)
  - Relationships between segments in pictures
  - ...

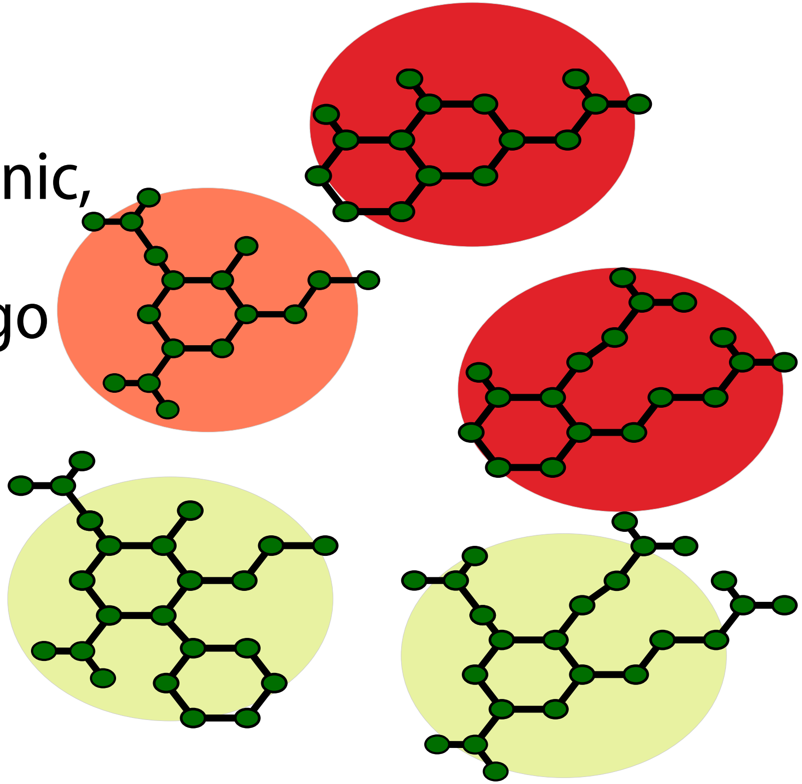






Tasks on graphs

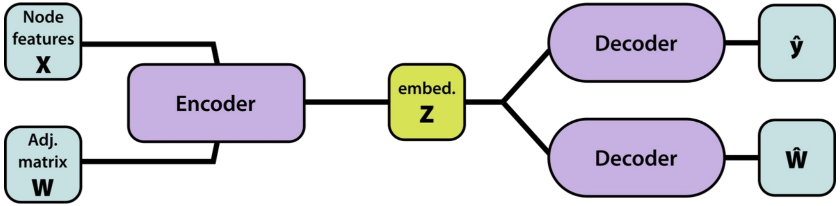
- Predict properties of graphs
  - Chemical properties (solubility, carcinogenic, possible drug)
  - Classification of the research field in an ego network
  - ...

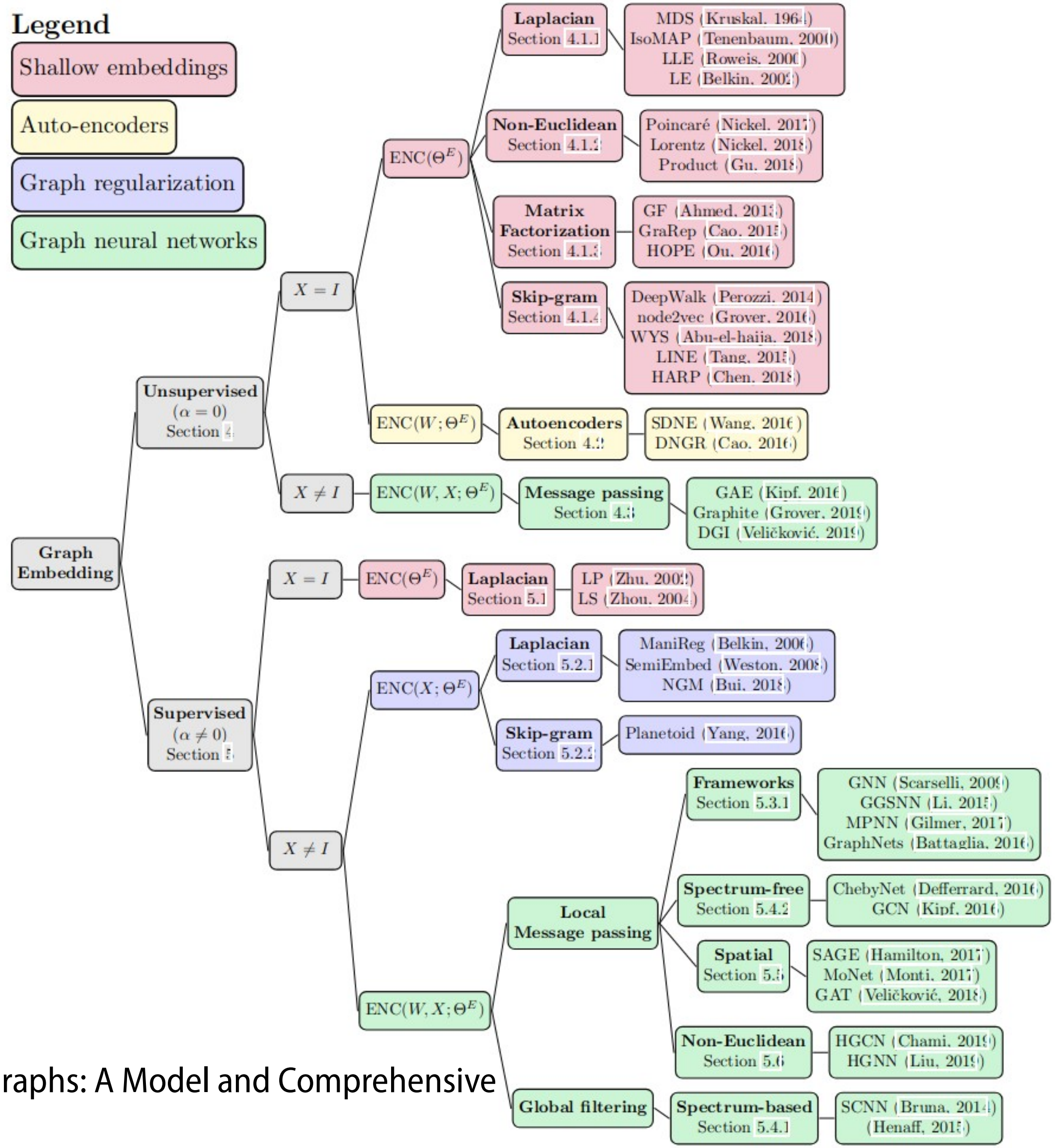


A few examples

FIDLE

Taxonomy of methods

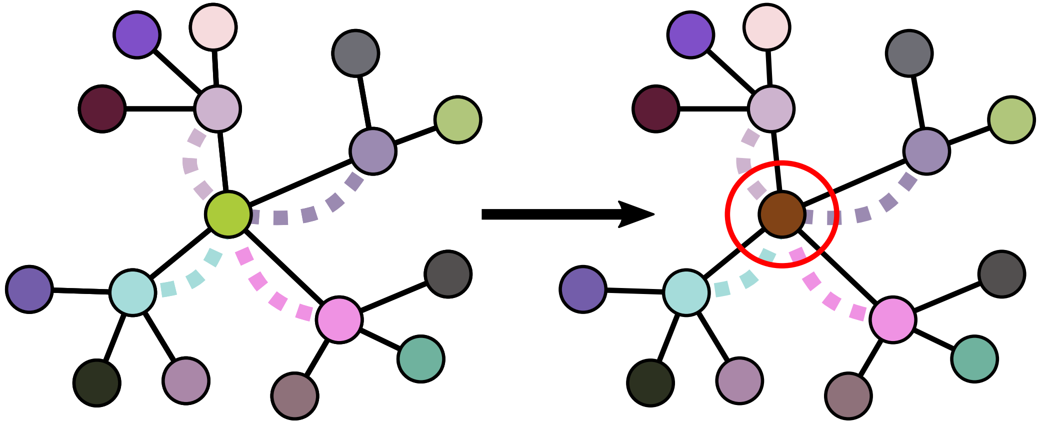




I. Chami, S. Abu-El-Haija, and B. Perozzi, "Machine Learning on Graphs: A Model and Comprehensive Taxonomy".

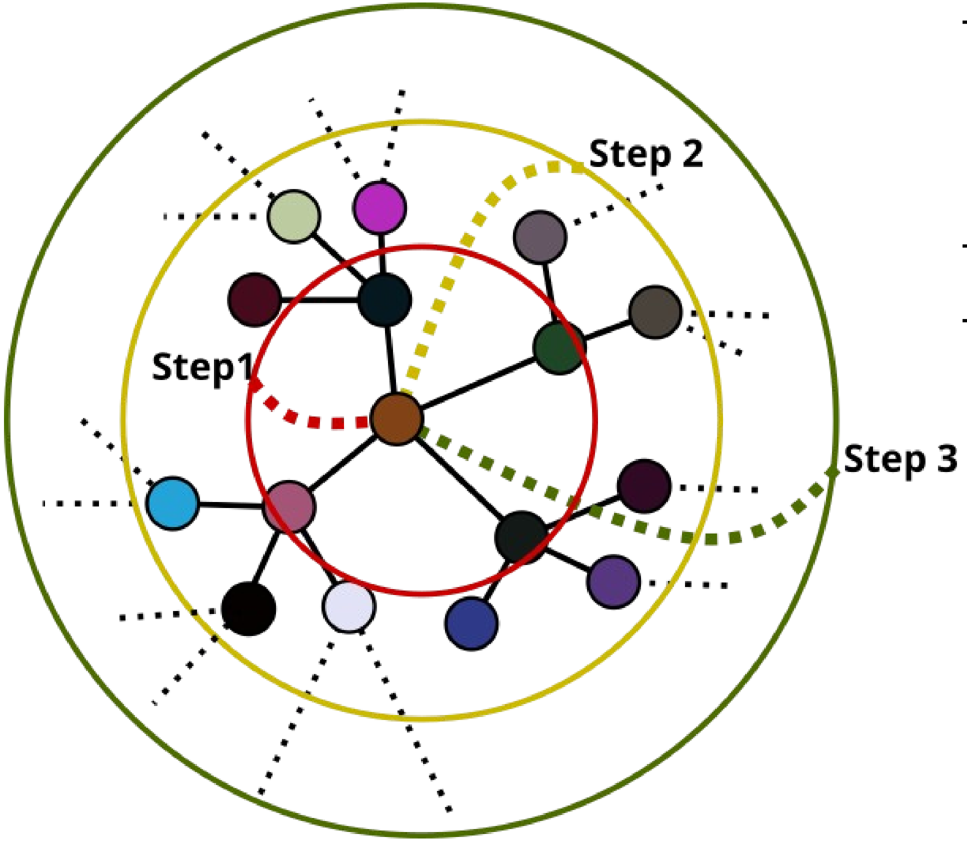
Graph convolution

- Just like for images we can learn from neighborhood with a convolution operator.



- A bit more complex since the number of neighbors is unlikely to be constant.
- We want the operator to be permutation invariant.

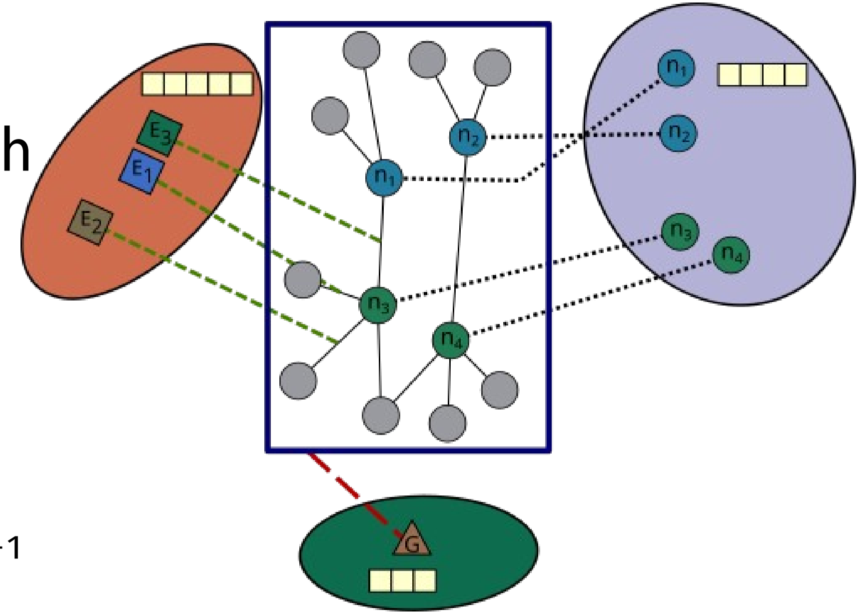
Graph convolution

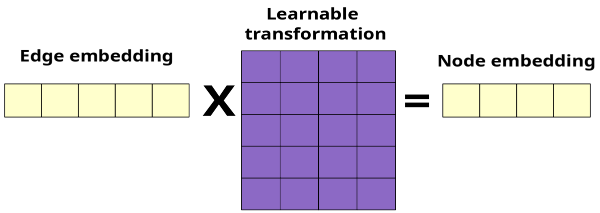


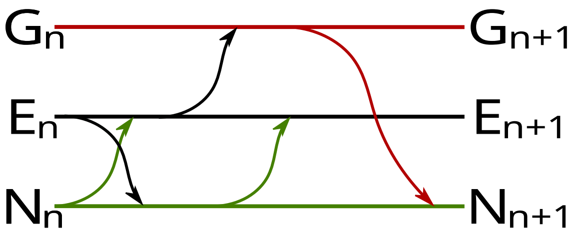
- Several steps are needed to retrieve information for distant nodes.
- For large graphs → a **cutoff**
- It is possible to use a **virtual node** connected to all other nodes. But in practice this becomes quickly intractable.

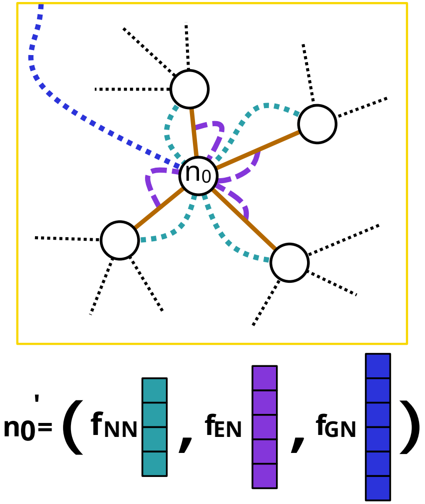
Message passing

- We have embeddings for each part of the graph (possibly different vector sizes).
- Each part can learn from the others via a transformation.



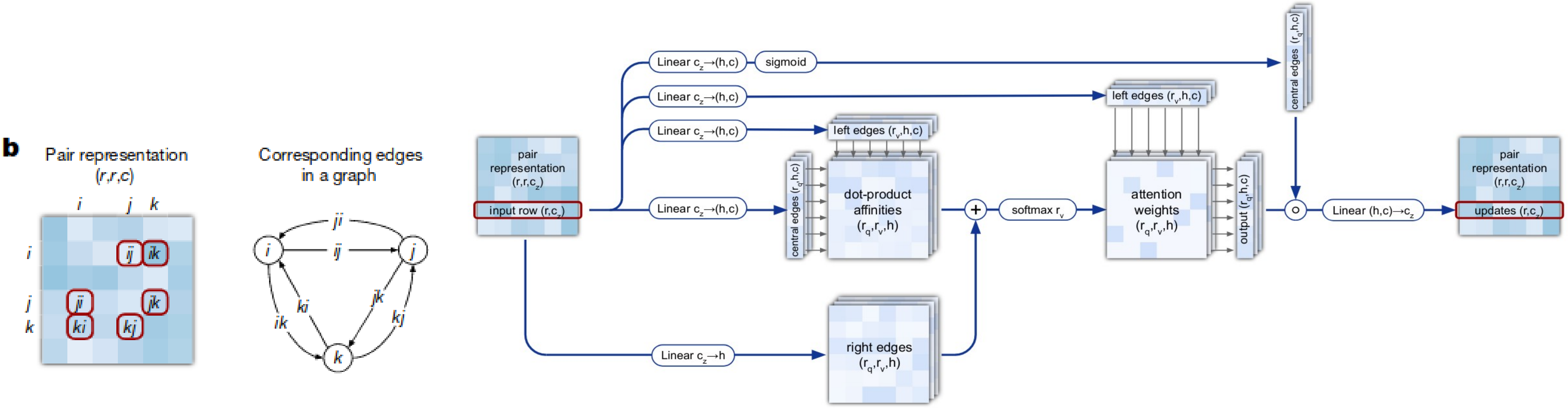






- Information is aggregated to form a message that the node/edge will send to others.

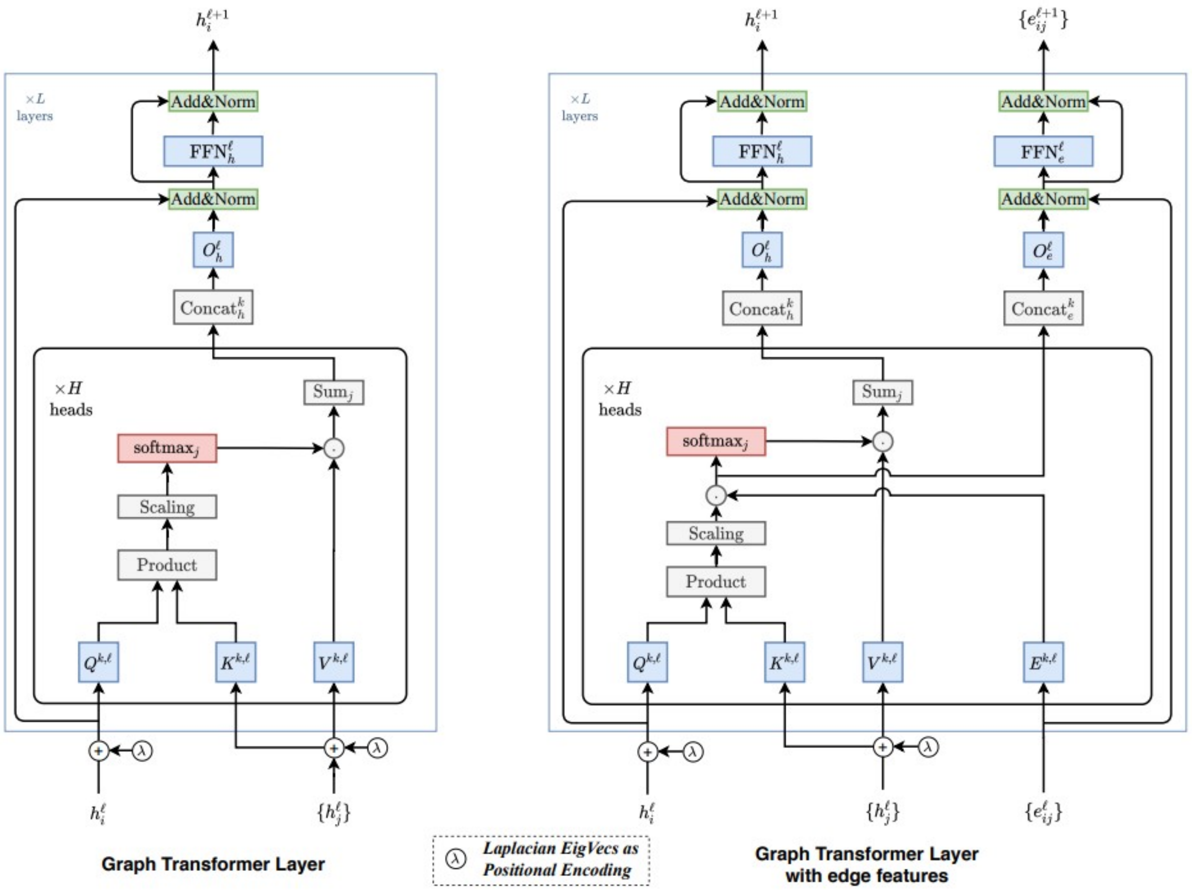
Alphafold transformer



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

J. Jumper et al., Highly accurate protein structure prediction with AlphaFold, Nature, vol. 596, no. 7873, Art. no. 7873, Aug. 2021, doi: 10.1038/s41586-021-03819-2.

Graph Transformer Network



Dwivedi, Bresson A Generalization of Transformer Networks to Graphs 2020, https://arxiv.org/abs/2012.09699

Resources

Libraries

- → Pytorch Geometric
- → Deep Graph Library
- → Graph Nets
- → Spektral
- → ...

- → https://logconference.org/
- → https://ogb.stanford.edu/

Tutorials

- → https://antoniolonga.github.io/Pytorch_geometric_tutorials/
- → https://docs.dgl.ai/tutorials/blitz

References

→ Books
  → Deep Learning on Graphs (Jiliang Tang and Yao Ma)
  → Introduction to Graph Neural Networks (Introduction to Graph Neural Networks)

→ Websites
  → https://distill.pub/2021/gnn-intro/
  → https://neptune.ai/blog/graph-neural-network-and-some-of-gnn-applications
  → https://venturebeat.com/2021/10/13/what-are-graph-neural-networks-gnn/
  → https://theaisummer.com/graph-convolutional-networks/
  → https://towardsdatascience.com/node-embeddings-for-beginners-554ab1625d98

→ Articles

- **Chami, S. Abu-El-Haija, and B. Perozzi, "Machine Learning on Graphs: A Model and Comprehensive Taxonomy".**
- Zhou, Jie, et al. "Graph neural networks: A review of methods and applications." AI Open 1 (2020): 57-81.
- Scarselli, Franco, et al. "The graph neural network model." IEEE transactions on neural networks 20.1 (2008): 61-80.
- Kipf, Thomas N., and Max Welling. "Semi-supervised classification with graph convolutional networks." arXiv preprint arXiv:1609.02907 (2016).
- Perozzi, Bryan, Rami Al-Rfou, and Steven Skiena. "Deepwalk: Online learning of social representations." Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. 2014.
- Shlomi, Jonathan, Peter Battaglia, and Jean-Roch Vlimant. "Graph neural networks in particle physics." Machine Learning: Science and Technology 2.2 (2020): 021001.
- Duong, Chi Thang, et al. "On node features for graph neural networks." arXiv preprint arXiv:1911.08795 (2019).
- Dwivedi, Bresson "A Generalization of Transformer Networks to Graphs" 2020, https://arxiv.org/abs/2012.09699