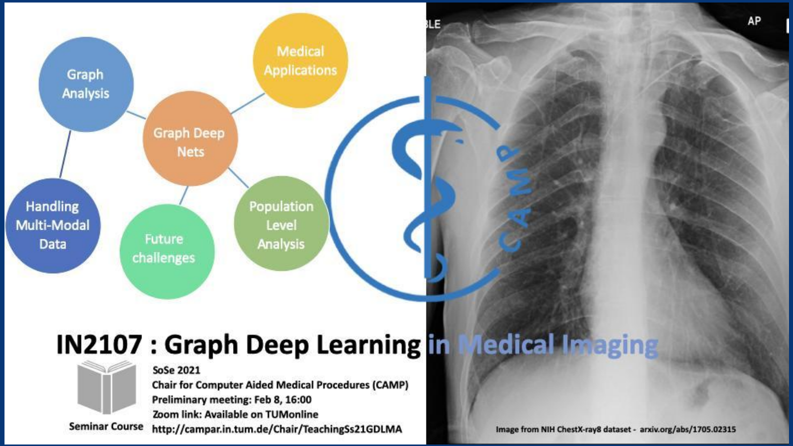
Chair for Computer Aided Medical Procedures (CAMP)
Master Seminar on
Graph Deep Learning for Medical Applications SoSe22



Dr. Anees Kazi, Mahsa Ghorbani, Kamilia Mullakaeva, Tamara Mueller, Matthias Keicher, Tobias Czempiel, Ege Özsoy, Felix Holm, Prof. Dr. Nassir Navab

Technische Universität München + Johns Hopkins Whiting School of Engineering

Basic Info about the course

- **Type**: Master Seminar (IN0014, IN2107, IN4431)
- **Language**: English
- **SWS**: 2
- **ECTS**: 5 Credits
- **Webpage**: https://wiki.tum.de/display/gdlma/GDLMA+SoSe22
- **Time**:
  - Tuesdays 12 pm to 2 pm
- **Location**:
  - Virtual Meeting Room (Zoom)
  - Seminar Room (03.09.012)
- **Requirements**:
  - Background in Machine/Deep Learning
    https://www.youtube.com/watch?v=Iiv9R6BjxHM

Objective

- Learn through **reading, understanding, presenting, and discussing**
- Challenges in Medical Applications with Graph Deep Learning:
  - Current state of the art
    - Node classification, graph classification, graph learning, population level data analysis
  - Future challenges
    - 3D data, local and global data analysis in data distribution, learning with incomplete data
  - Handling Multi-Modal Data
  - Robustness and interpretability for GCN
  - Application to Medical imaging
  - Understanding the mathematical details

Course Evaluation

Presentation (45%)

- 20 minutes + 10 minutes Q&A
- Slides (Powerpoint, Latex, see website for templates)
- They should cover all relevant aspects of the paper
  - Motivation
  - Methodology
  - Experimental results
  - Take Home Message
  - Discussion
- Self-contained (review of state of the art is necessary!)
- Presentation guidelines will be released later.
- **All students are expected to attend all presentations and interact during Q&A**
- **Examples from previous semester:**

  https://wiki.tum.de/display/dlma/Presentations%3A+Summer+2020

Blog Post (45%)

- Blog post explaining the main ideas of the paper.
  - Motivation + Contributions
  - Methodology
  - Results & Discussion
- You can refer to https://bair.berkeley.edu/blog/ to get ideas
- 1500-1700 words paper summary + 200-300 words your own review
- Students will be requested to comment on each other's blog posts.
- The website where the posts will be uploaded is [1].
- You can later privately share your blog posts in other websites as well (eg Medium).
- Upload the blog post two weeks before presentation. There will be discussion until presentation
- **Examples from previous semester:**

  https://wiki.tum.de/display/dlma/Blog%3A+Summer+2020

Attendance and Participation (10%)

Schedule

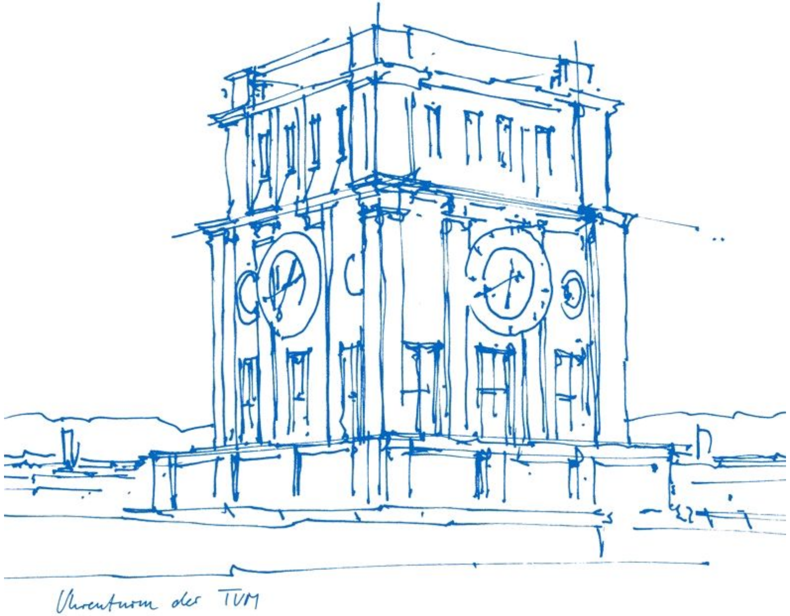
Sessions will take place **Tuesdays from 12:00 to 14:00h**. A group of Two to Three students will present their papers during each session.

Date	Time	Place	Topic
26.04.22	12:00 - 14:00	03.09.012	Lecture: Intro to GDLMA
03.05.22	12:00 - 14:00	03.09.012	Paper session
10.05.22	12:00 - 14:00	03.09.012	Paper session
17.05.22	12:00 - 14:00	03.09.012	Talk 1: Dr. Seyed Ahmad Ahmadi (NVIDIA)
24.05.22	12:00 - 14:00	03.09.012	Paper session
31.05.22	12:00 - 14:00	03.09.012	Paper session
14.06.22	12:00 - 14:00	03.09.012	Talk 2 : Alaa Bessadok (Helmholtz)
21.06.22	12:00 - 14:00	03.09.012	Paper session
28.06.22	12:00 - 14:00	03.09.012	Lecture: Intro to Scene Graphs
5.07.22	12:00 - 14:00	03.09.012	Paper session
12.07.22	12:00 - 14:00	03.09.012	Paper session
19.07.22	12:00 - 14:00	03.09.012	Talk 3 : Azade Farshad (CAMP)
26.07.22	12:00 - 14:00	03.09.012	Paper session - Backup

Paper assignments: https://wiki.tum.de/display/gdlma/GDLMA+SoSe22

Graph Deep Learning for Medical Applications

Lecture I



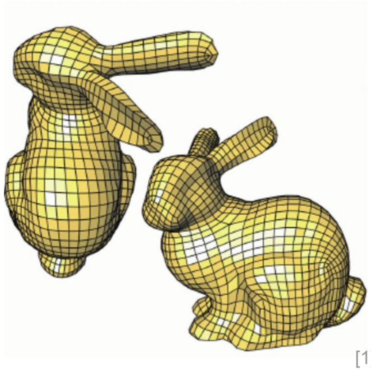
26.04.2022

Geometric Deep Learning (GDL)

"Geometric deep learning is an
umbrella term
for emerging techniques attempting to
generalize
(structured) ***deep neural models***
to non-Euclidean domains*,*
such as graphs and manifolds."

Bronstein, Michael M., et al. "Geometric deep learning: going beyond euclidean data." *IEEE Signal Processing Magazine* 34.4 (2017): 18-42.

Non-Euclidean Spaces

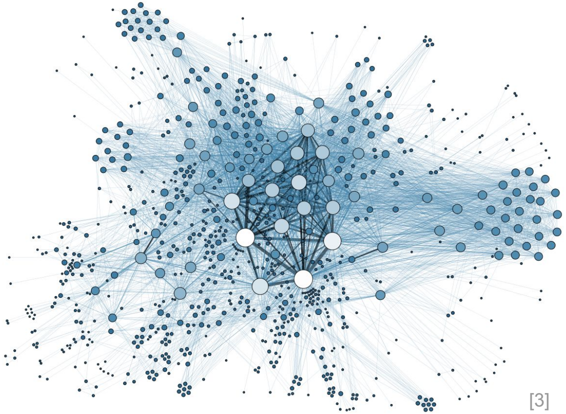


[1]

Manifolds / Surfaces

[2]

Point Clouds



[3]

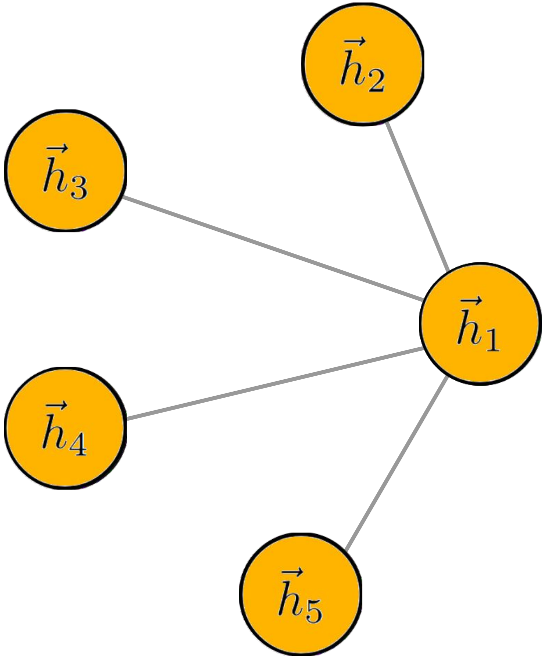
Graphs

[1] https://www.researchgate.net/figure/Remeshing-the-Stanford-Bunny-Left-Pure-quad-remeshing-Right-detail-of-a-half-pole_fig9_221316594
[2] Yang, Guandao, et al. "Pointflow: 3d point cloud generation with continuous normalizing flows." (2019).
[3] https://commons.wikimedia.org/wiki/File:Social_Network_Analysis_Visualization.png

Graphs

$$G = (V, E)$$

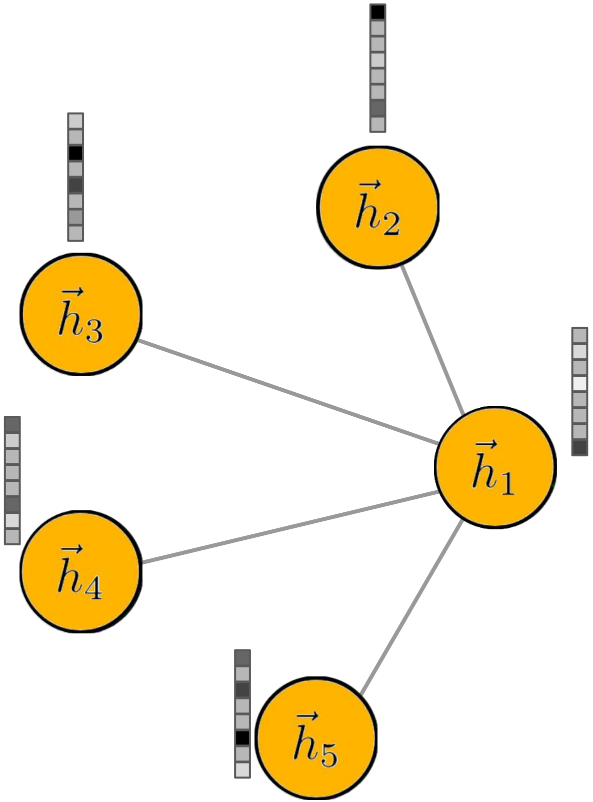
- Set of **nodes** (arbitrary ordering) and **edges**



Graphs

$$G = (V, E)$$

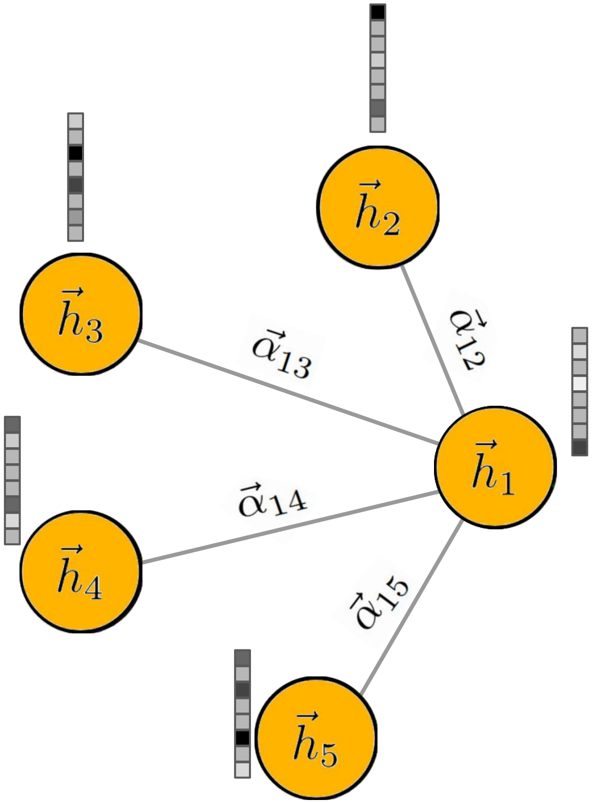
- Set of **nodes** (arbitrary ordering) and **edges**
- Feature vector for each node



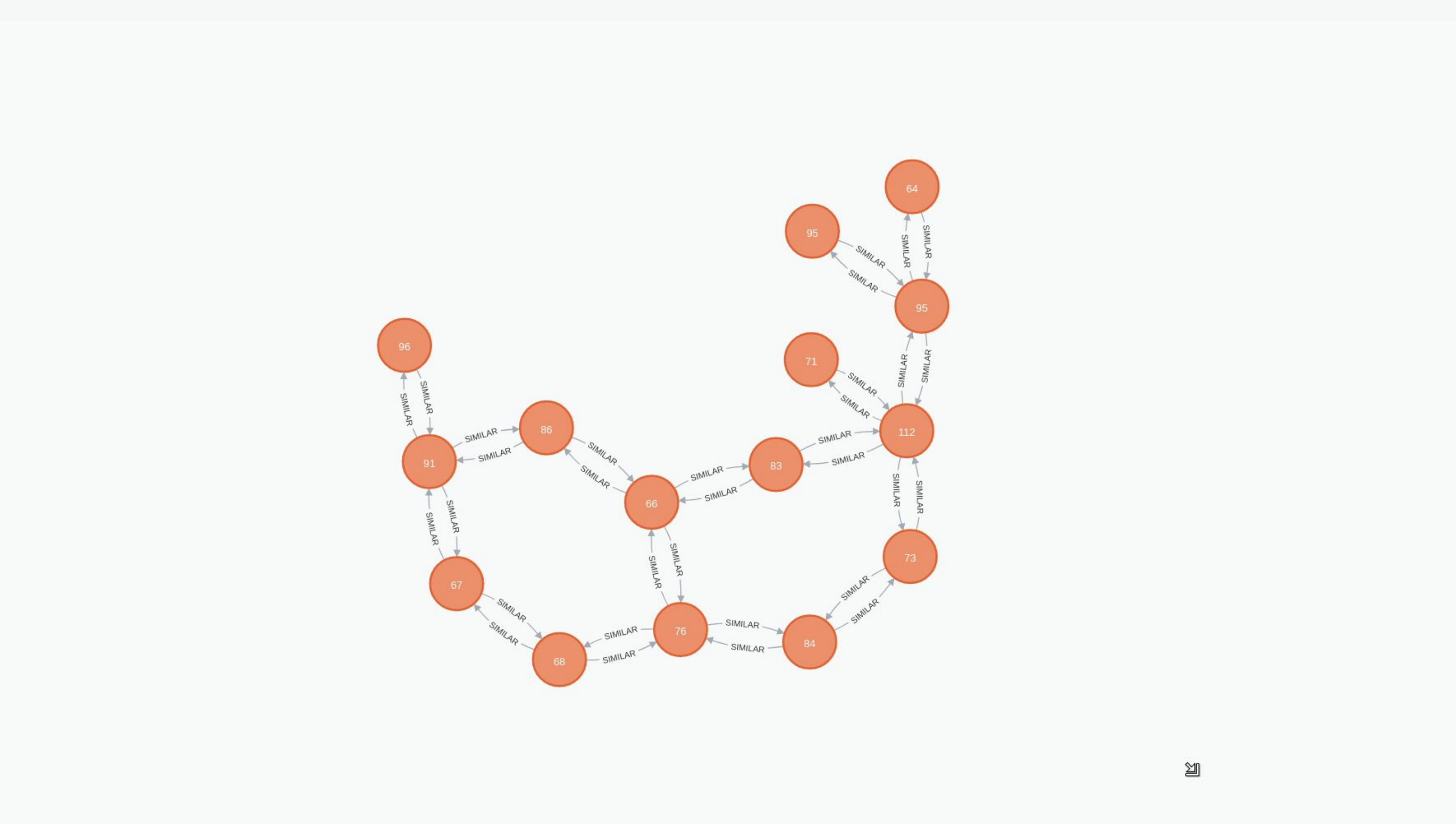
Graphs

$$G = (V, E)$$

- Set of **nodes** (arbitrary ordering) and **edges**
- Feature vector for each node
- Edge attributes



Graphs



Convolutional Networks

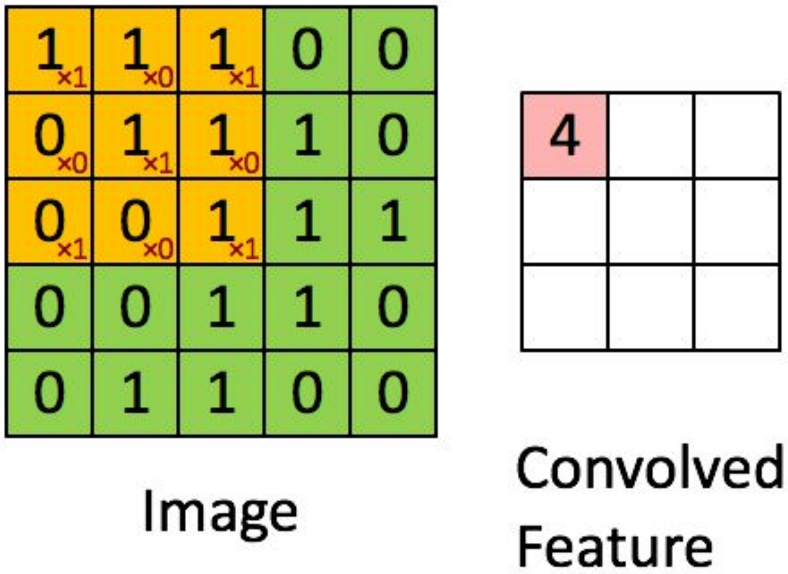
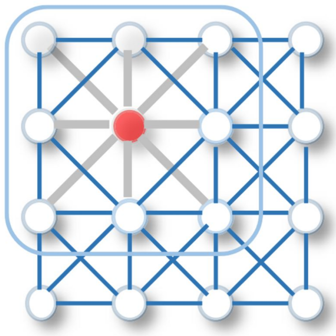
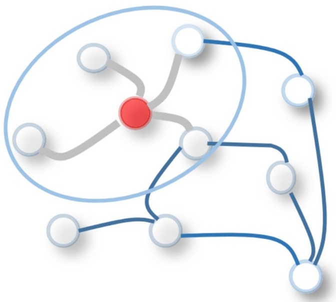


Fig.: https://algorithmia.com/blog/convolutional-neural-nets-in-pytorch

Images vs. Graphs





Images

- well-structured, grid-based
- Fixed number of neighbours
- Fixed position

→ **Convolution using Filters**

Graphs

- Flexible data structure
- Varying number of neighbours
- Flexible position of nodes

→ **Graph Convolution**

Wu, Zonghan, et al. "A comprehensive survey on graph neural networks." (2020).

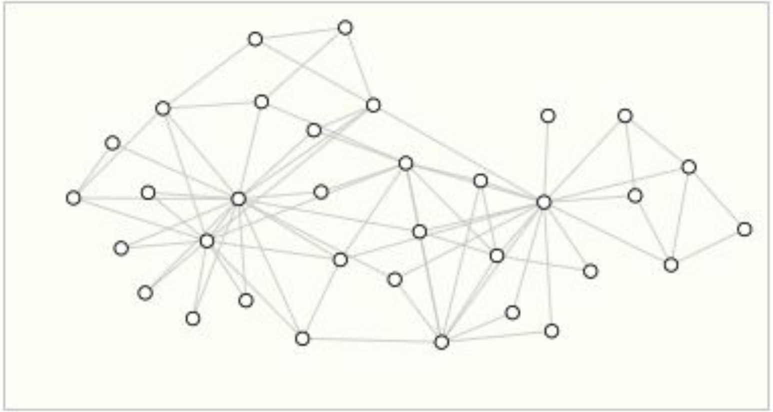
Graph Neural Networks

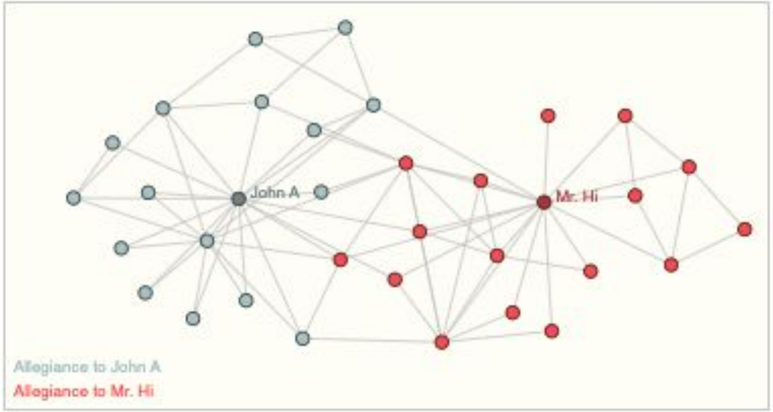
Learning directly on graph-structured data

Graph Neural Network Tasks

Node Classification/Regression

Determine a label for each node.





Input graph: unlabeled or partially labeled nodes

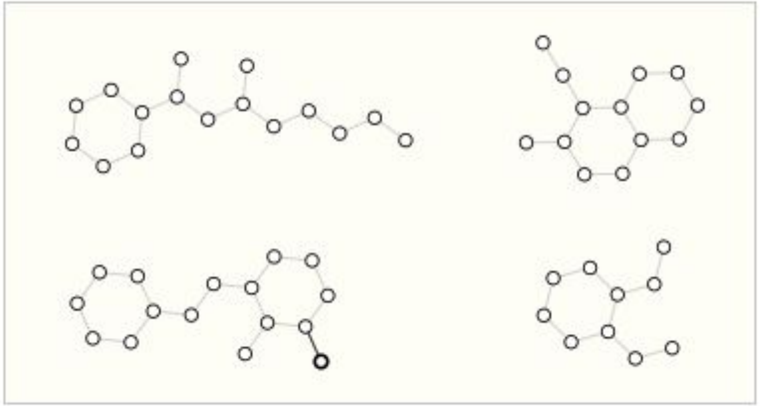
Output graph: labeled nodes

https://distill.pub/2021/gnn-intro/

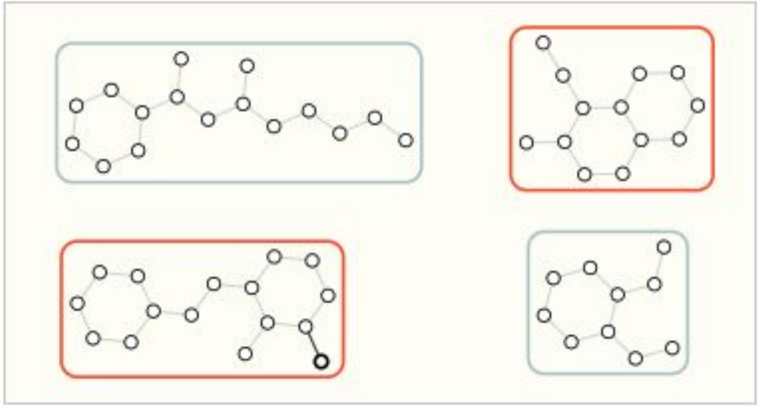
Graph Neural Network Tasks

Graph Classification/Regression

Classify the whole graph into different categories.



Input: set of graphs



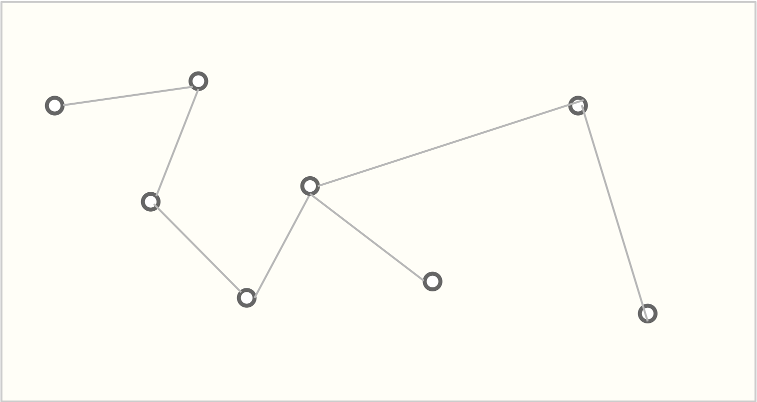
Output: labels for each graph, (e.g., "does the graph contain two rings?")

https://distill.pub/2021/gnn-intro/

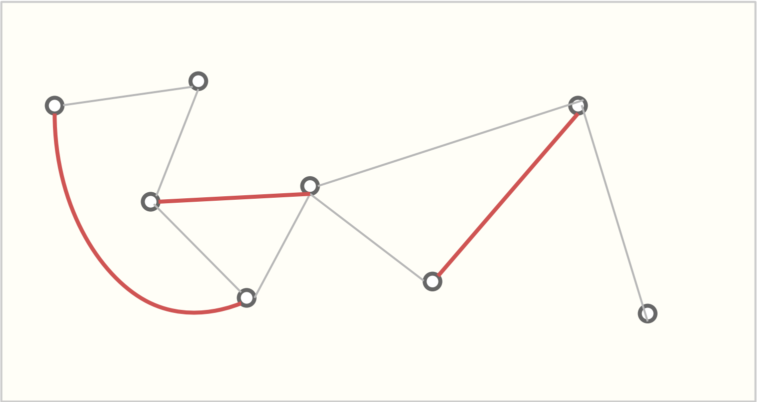
Graph Neural Network Tasks

Link Prediction

Predict whether an edge exists between two given nodes or not.



Input: Incomplete graph



Output: Graph with predicted edges

https://distill.pub/2021/gnn-intro/

Transductive vs. Inductive Learning

Transduction:

→ *Reasoning from observed, specific (training) cases to specific (test) cases*

- All data is observed beforehand (training and testing)
- Labels of the test data is unknown
- But the test data points are used during the learning process

→ Does not work for new/unseen data

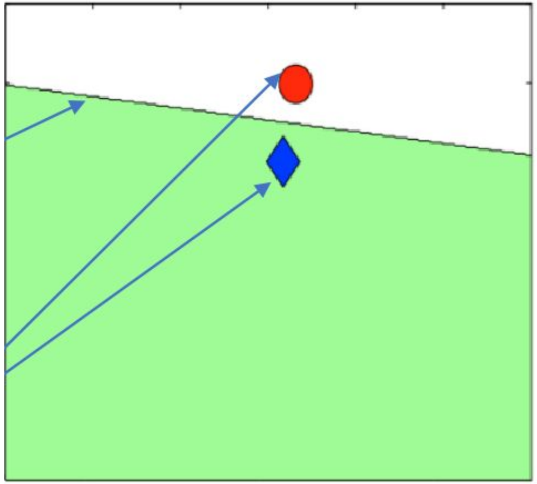
Induction:

→ *Reasoning from observed (training) data to general rules, which are then applied to test cases*

- Equivalent to supervised learning
- Only training data is used during learning
- Then the learned rules are applied to new unseen (test) data

→ Also works for new/unseen data

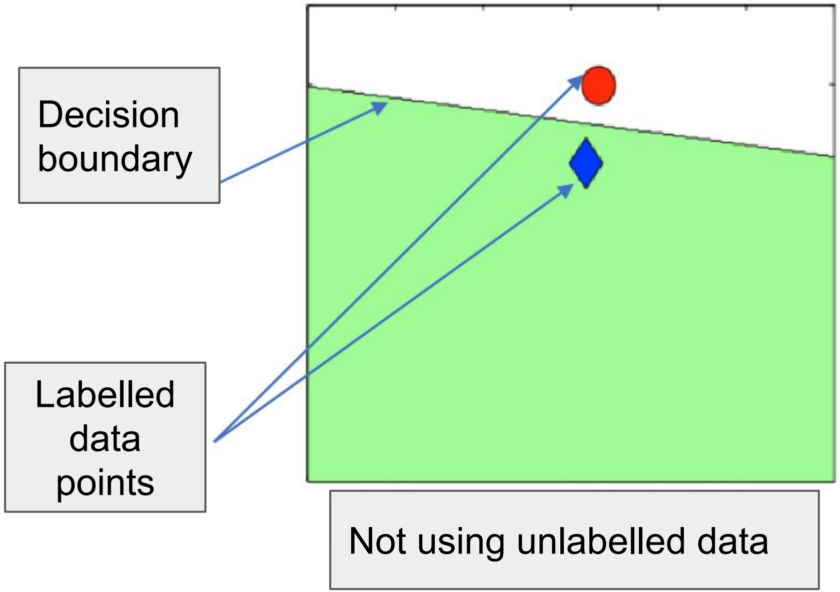
Transductive vs. Inductive Learning



Example from [Belkin et al., JMLR 2006]

Figure adapted from lecture "*Machine Learning for Graphs and Sequential Data*" *2021*

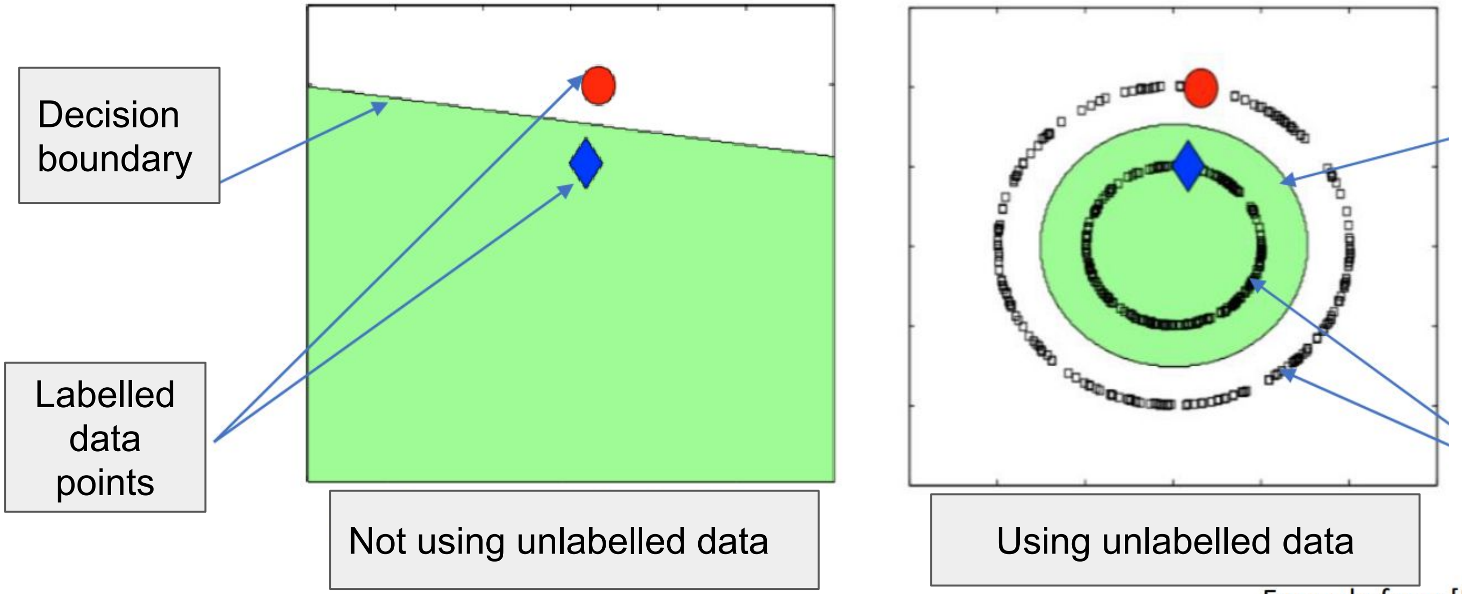
Transductive vs. Inductive Learning



Example from [Belkin et al., JMLR 2006]

Figure adapted from lecture "*Machine Learning for Graphs and Sequential Data*" *2021*

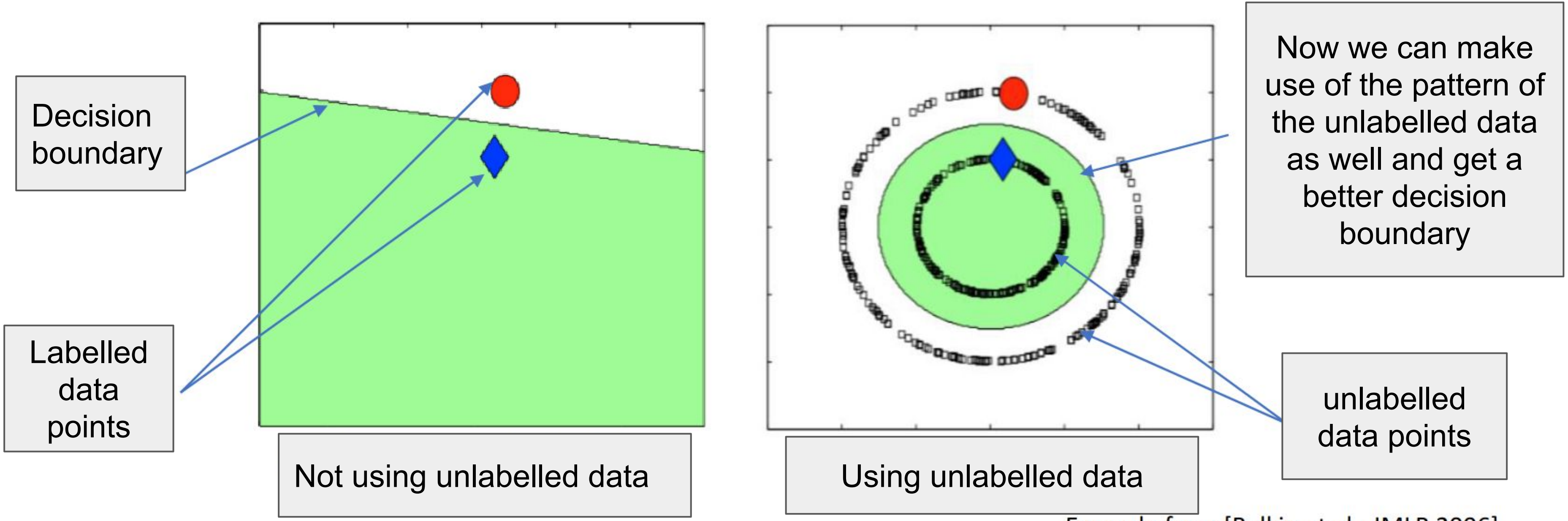
Transductive vs. Inductive Learning



Example from [Belkin et al., JMLR 2006]

Figure adapted from lecture “*Machine Learning for Graphs and Sequential Data*” *2021*

Transductive vs. Inductive Learning



Example from [Belkin et al., JMLR 2006]

Spectral vs. Spatial GNNs

Spectral

- The method works in the **eigenvalue domain** of the feature representation
- The eigenvalue domain is defined by the **complete dataset** (a local structure in the graph affects the global representation)

→ Extension to unseed data not easily possible

→ Transductive Learning

Spatial

- Using the node neighbourhood
- Instead of the eigenvalue domain

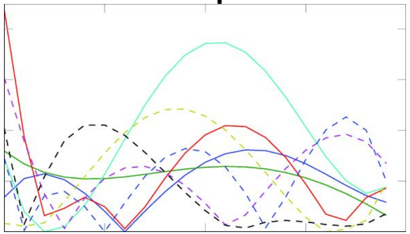
→ An extension to new samples is possible **without retraining the network**

→ Inductive (and Transductive) Learning

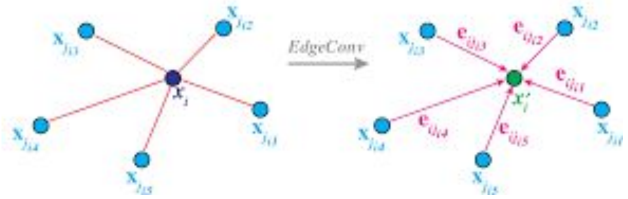
Spectral vs. Spatial GNNs

GCN

Spectral

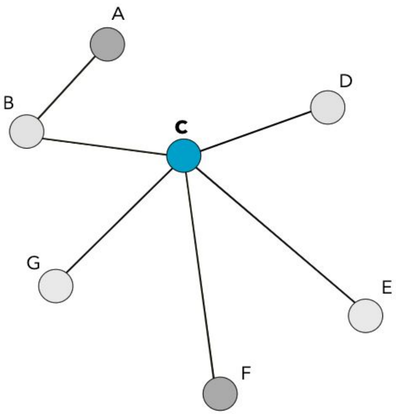


Spatial



Spectral Graph Convolutions

Graph



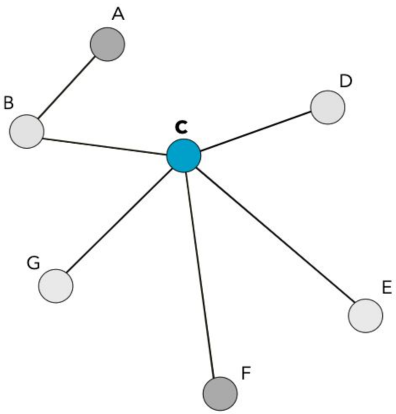
Adjacency Matrix

$$
\begin{array}{c c}
 & \begin{array}{ccccccc} A & B & \mathbf{C} & D & E & F & G \end{array} \\
\begin{array}{c} A \\ B \\ \mathbf{C} \\ D \\ E \\ F \\ G \end{array} &
\begin{bmatrix}
0 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 1 & 1 & 1 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0
\end{bmatrix}
\end{array}
$$

https://distill.pub/2021/understanding-gnns/

Spectral Graph Convolutions

Graph



$$\begin{array}{c|ccccccc} & A & B & \mathbf{C} & D & E & F & G \\ \hline A & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ B & 0 & 2 & 0 & 0 & 0 & 0 & 0 \\ \mathbf{C} & 0 & 0 & 5 & 0 & 0 & 0 & 0 \\ D & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ E & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ F & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ G & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array}$$

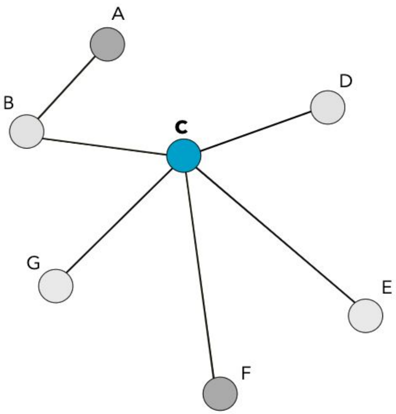
$$\boxed{D_v = \sum_u A_{vu}.}$$

The degree of node v is the number of edges incident at v.

Degree Matrix

https://distill.pub/2021/understanding-gnns/

Spectral Graph Convolutions



Laplacian

$$\boxed{L = D - A}$$

L = Laplacian of the Graph
D = Degree Matrix of the Graph
A = Adjacency Matrix of the Graph

$$
\begin{array}{c}
 \\ \text{A} \\ \text{B} \\ \textbf{C} \\ \text{D} \\ \text{E} \\ \text{F} \\ \text{G}
\end{array}
\begin{array}{c}
\begin{array}{ccccccc} \text{A} & \text{B} & \textbf{C} & \text{D} & \text{E} & \text{F} & \text{G} \end{array} \\
\begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 2 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 5 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
\end{array}
-
\begin{array}{c}
 \\ \text{A} \\ \text{B} \\ \textbf{C} \\ \text{D} \\ \text{E} \\ \text{F} \\ \text{G}
\end{array}
\begin{array}{c}
\begin{array}{ccccccc} \text{A} & \text{B} & \textbf{C} & \text{D} & \text{E} & \text{F} & \text{G} \end{array} \\
\begin{bmatrix}
0 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 1 & 1 & 1 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0
\end{bmatrix}
\end{array}
=
\begin{array}{c}
 \\ \text{A} \\ \text{B} \\ \textbf{C} \\ \text{D} \\ \text{E} \\ \text{F} \\ \text{G}
\end{array}
\begin{array}{c}
\begin{array}{ccccccc} \text{A} & \text{B} & \textbf{C} & \text{D} & \text{E} & \text{F} & \text{G} \end{array} \\
\begin{bmatrix}
1 & -1 & & & & & \\
-1 & 2 & -1 & & & & \\
 & -1 & 5 & -1 & -1 & -1 & -1 \\
 & & -1 & 1 & & & \\
 & & -1 & & 1 & & \\
 & & -1 & & & 1 & \\
 & & -1 & & & & 1
\end{bmatrix}
\end{array}
$$

https://distill.pub/2021/understanding-gnns/

Spectral Graph Convolutions

- Laplacian is a **real**, **symmetric** matrix, which means it has all real eigenvalues
- The set of eigenvalues of the Laplacian is called its *'spectrum'*

→ Spectral Graph Convolutions

Polynomials of the Laplacian :

$$p_w(L) = w_0 I_n + w_1 L + w_2 L^2 + \ldots + w_d L^d = \sum_{i=0}^{d} w_i L^i.$$

These polynomials can be thought of as the equivalent of 'filters' in CNNs, and the coefficients was the weights of the 'filters'.

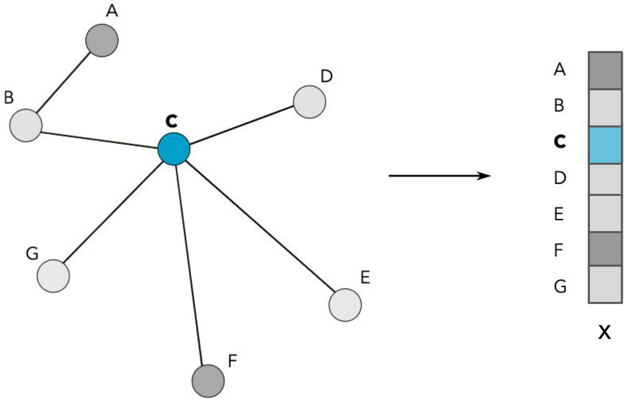
https://distill.pub/2021/understanding-gnns/

Spectral Graph Convolutions

Polynomials of the Laplacian :

$$p_w(L) = w_0 I_n + w_1 L + w_2 L^2 + \ldots + w_d L^d = \sum_{i=0}^{d} w_i L^i.$$

These polynomials can be thought of as the equivalent of ‘filters’ in CNNs, and the coefficients was the weights of the ‘filters’.



Resulting convolution: $\boxed{x' = p_w(L)\, x}$

https://distill.pub/2021/understanding-gnns/

Spectral Graph Convolutions

$$p_w(L) = w_0 I_n + w_1 L + w_2 L^2 + \ldots + w_d L^d = \sum_{i=0}^{d} w_i L^i.$$

Let's look at the simplest convolution where $w_0 = 1$ and all of the other coefficients are 0:

$$x' = p_w(L)\, x = \sum_{i=0}^{d} w_i L^i x = w_0 I_n x = x.$$

https://distill.pub/2021/understanding-gnns/

Spectral Graph Convolutions

$$p_w(L) = w_0 I_n + w_1 L + w_2 L^2 + \ldots + w_d L^d = \sum_{i=0}^{d} w_i L^i.$$

Let's look at the simplest convolution where $w_1 = 1$ and all of the other coefficients are 0:

$$\begin{aligned} x'_v = (Lx)_v &= L_v x \\ &= \sum_{u \in G} L_{vu} x_u \\ &= \sum_{u \in G} (D_{vu} - A_{vu}) x_u \\ &= D_v \, x_v - \sum_{u \in \mathcal{N}(v)} x_u \end{aligned}$$

https://distill.pub/2021/understanding-gnns/

Spectral Graph Convolutions

$$p_w(L) = w_0 I_n + w_1 L + w_2 L^2 + \ldots + w_d L^d = \sum_{i=0}^{d} w_i L^i.$$

Let's look at the simplest convolution where $w_1 = 1$ and all of the other coefficients are 0:

$$\begin{aligned} x'_v = (Lx)_v &= L_v x \\ &= \sum_{u \in G} L_{vu} x_u \\ &= \sum_{u \in G} (D_{vu} - A_{vu}) x_u \\ &= D_v \, x_v - \sum_{u \in \mathcal{N}(v)} x_u \end{aligned}$$

x_v is updated with its own features…

https://distill.pub/2021/understanding-gnns/

Spectral Graph Convolutions

$$p_w(L) = w_0 I_n + w_1 L + w_2 L^2 + \ldots + w_d L^d = \sum_{i=0}^{d} w_i L^i.$$

Let's look at the simplest convolution where $w_1 = 1$ and all of the other coefficients are 0:

$$\begin{aligned} x'_v = (Lx)_v &= L_v x \\ &= \sum_{u \in G} L_{vu} x_u \\ &= \sum_{u \in G} (D_{vu} - A_{vu}) x_u \\ &= D_v \, x_v - \sum_{u \in \mathcal{N}(v)} x_u \end{aligned}$$

x_v is updated with its own features…

… and with the features from its immediate neighbours x_u

https://distill.pub/2021/understanding-gnns/

Spectral Graph Convolutions

$$p_w(L) = w_0 I_n + w_1 L + w_2 L^2 + \ldots + w_d L^d = \sum_{i=0}^{d} w_i L^i.$$

Let's look at the simplest convolution where $w_1 = 1$ and all of the other coefficients are 0:

$$\begin{aligned} x'_v = (Lx)_v &= L_v x \\ &= \sum_{u \in G} L_{vu} x_u \\ &= \sum_{u \in G} (D_{vu} - A_{vu}) x_u \\ &= D_v\, x_v - \sum_{u \in \mathcal{N}(v)} x_u \end{aligned}$$

x_v is updated with its own features…

… and with the features from its immediate neighbours x_u

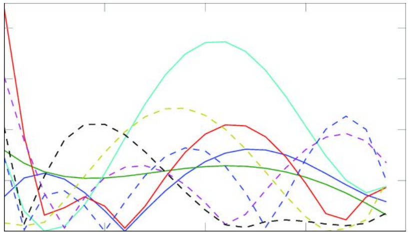
'Message passing' with 1-hop localized convolution.

https://distill.pub/2021/understanding-gnns/

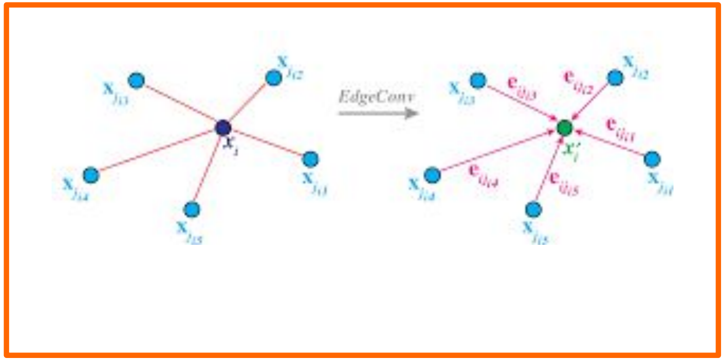
Spectral vs. Spatial GNNs

GCN

Spectral

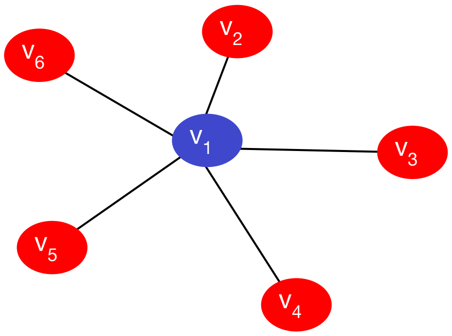


Spatial



General idea of inductive graph approach

General idea for spatial GCN:
Feature representation of one node gets updated by surrounding nodes

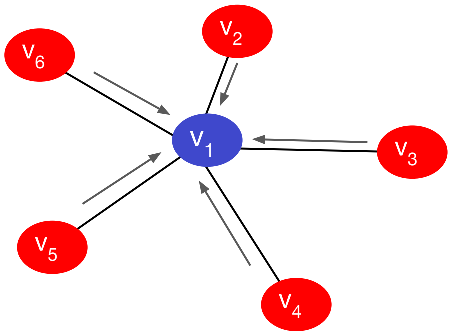


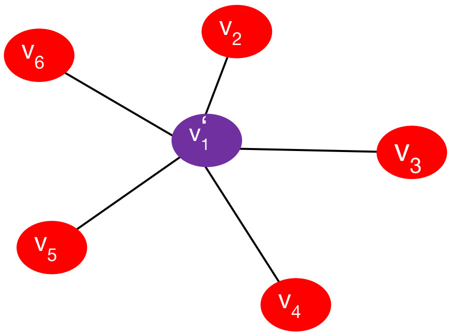
Node 1 with feature representation v_1 has connections to five corresponding neighbors

General idea of inductive graph approach

General idea for spatial GCN:

Feature representation of one node gets updated by surrounding nodes





Node 1 with feature representation v_1 has connections to five corresponding neighbors

Surrounding nodes features are aggregated and result in new representation v'_1

Graph Convolutions

Spatial GCN:

$$h_v^{(k)} = f^{(k)}\left(W^{(k)} \cdot \frac{\sum\limits_{u \in \mathcal{N}(v)} h_u^{(k-1)}}{|\mathcal{N}(v)|} + B^{(k)} \cdot h_v^{(k-1)}\right) \quad \text{for all } v \in V.$$

Node v's embedding at step k.

Mean of v's neighbour's embeddings at step $k-1$.

Node v's embedding at step $k-1$.

Color Codes:

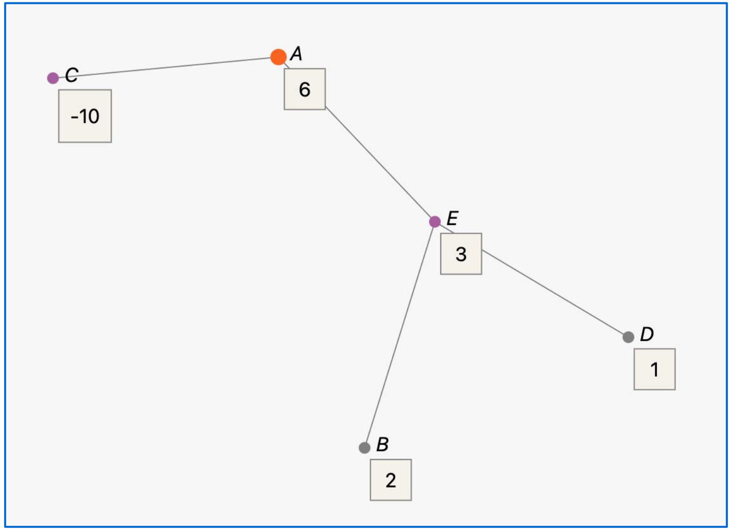
- Embedding of node v.
- Embedding of a neighbour of node v.
- (Potentially) Learnable parameters.

https://distill.pub/2021/understanding-gnns/

Kipf, Welling. "Semi-Supervised Classification with Graph Convolutional Networks", 2016.

Graph Convolutions

Practical example on spatial GCN:



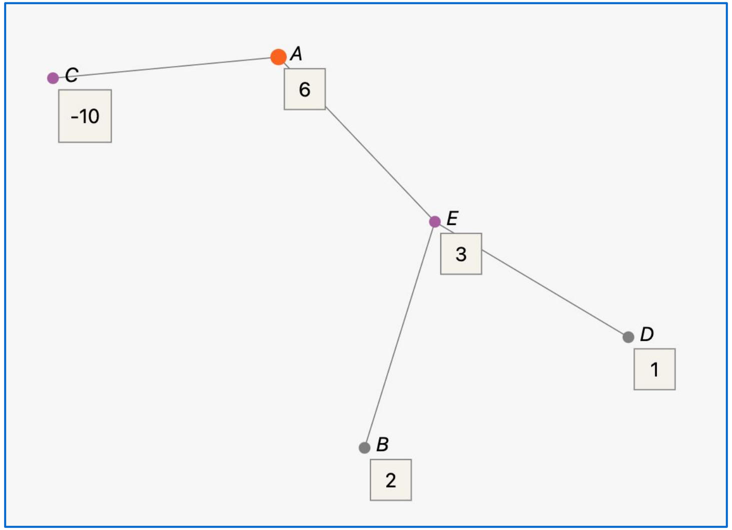
$$W^{(1)} = 1, B^{(1)} = 1$$

$$\begin{aligned} h_A^{(1)} &= f\left(W^{(1)} \times \frac{h_C^{(0)} + h_E^{(0)}}{2} + B^{(1)} \times h_A^{(0)}\right) \\ &= f\left(1 \times \frac{-10+3}{2} + 1 \times 6\right) \\ &= f(-3.5 + 6) \\ &= f(2.5) \\ &= ReLU(2.5) = 2.5 \end{aligned}$$

https://distill.pub/2021/understanding-gnns/

Graph Convolutions

Practical example on spatial GCN:



$$h_v^{(k)} = f^{(k)}\left(W^{(k)} \cdot \frac{\sum_{u \in \mathcal{N}(v)} h_u^{(k-1)}}{|\mathcal{N}(v)|} + B^{(k)} \cdot h_v^{(k-1)}\right)$$

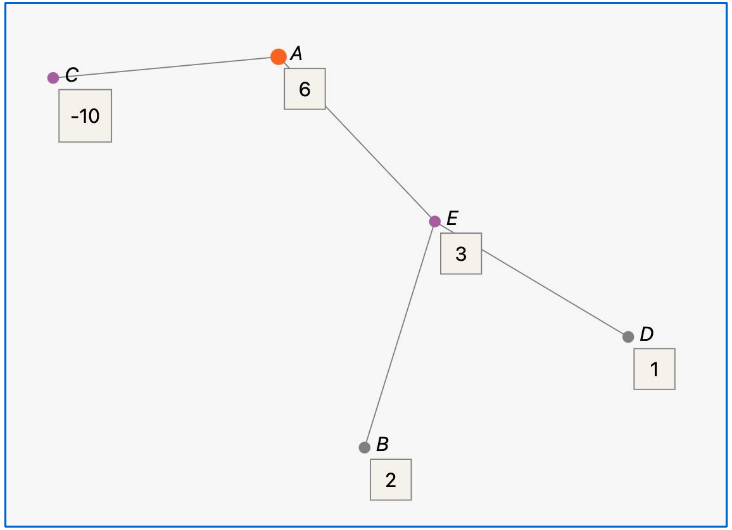
$$h_E^{(1)} =$$

$$W^{(1)} = 1, B^{(1)} = 1$$

https://distill.pub/2021/understanding-gnns/

Graph Convolutions

Practical example on spatial GCN:



$$h_v^{(k)} = f^{(k)}\left(W^{(k)} \cdot \frac{\sum_{u \in \mathcal{N}(v)} h_u^{(k-1)}}{|\mathcal{N}(v)|} + B^{(k)} \cdot h_v^{(k-1)}\right)$$

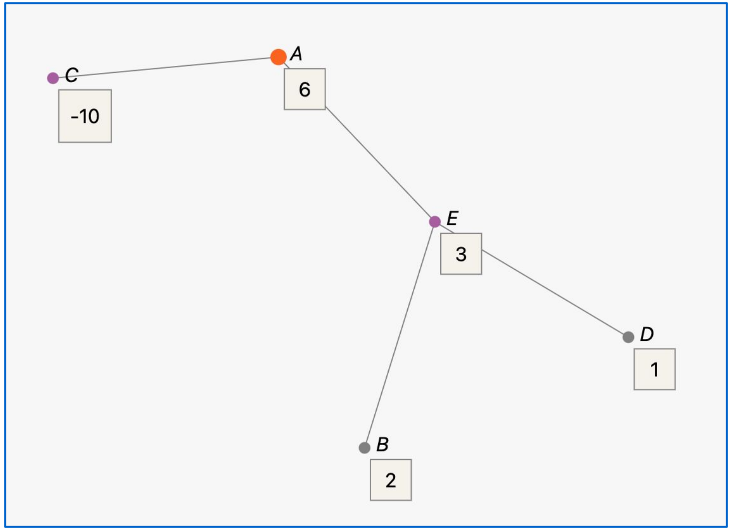
$$h_E^{(1)} = f(W^{(1)} \times \frac{h_A^{(0)} + h_B^{(0)} + h_D^{(0)}}{3} + B^{(1)} \times h_E^{(0)})$$

$$W^{(1)} = 1, B^{(1)} = 1$$

https://distill.pub/2021/understanding-gnns/

Graph Convolutions

Practical example on spatial GCN:



$$h_v^{(k)} = f^{(k)}\left(W^{(k)} \cdot \frac{\sum_{u \in \mathcal{N}(v)} h_u^{(k-1)}}{|\mathcal{N}(v)|} + B^{(k)} \cdot h_v^{(k-1)}\right)$$

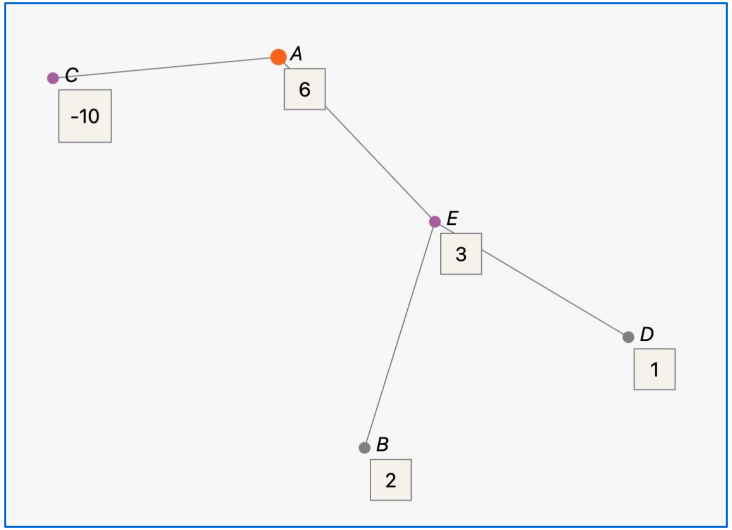
$$h_E^{(1)} = f\left(W^{(1)} \times \frac{h_A^{(0)} + h_B^{(0)} + h_D^{(0)}}{3} + B^{(1)} \times h_E^{(0)}\right)$$
$$= f\left(1 \times \frac{6+2+1}{3} + 1 \times 3\right)$$

$$W^{(1)} = 1, B^{(1)} = 1$$

https://distill.pub/2021/understanding-gnns/

Graph Convolutions

Practical example on spatial GCN:



$$W^{(1)} = 1, B^{(1)} = 1$$

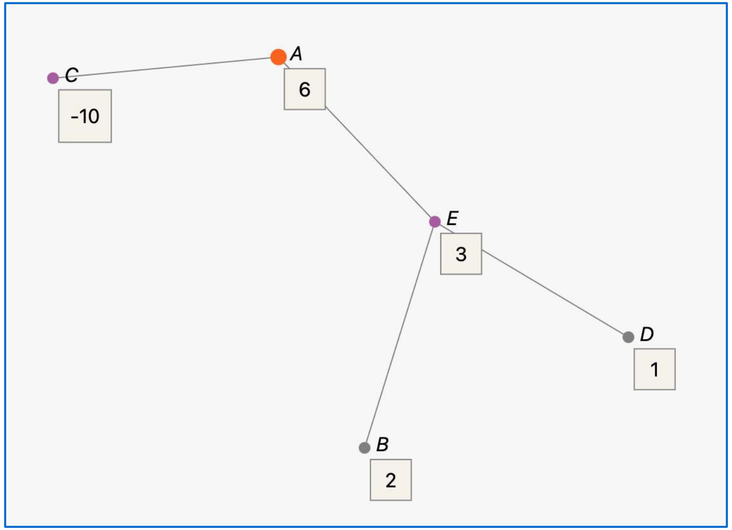
$$\begin{aligned} h_A^{(1)} &= f\left(W^{(1)} \times \frac{h_C^{(0)} + h_E^{(0)}}{2} + B^{(1)} \times h_A^{(0)}\right) \\ &= f\left(1 \times \frac{-10+3}{2} + 1 \times 6\right) \\ &= f(-3.5+6) \\ &= f(2.5) \\ &= ReLU(2.5) = 2.5 \end{aligned}$$

$$\begin{aligned} h_E^{(1)} &= f\left(W^{(1)} \times \frac{h_A^{(0)} + h_B^{(0)} + h_D^{(0)}}{3} + B^{(1)} \times h_E^{(0)}\right) \\ &= f\left(1 \times \frac{6+2+1}{3} + 1 \times 3\right) \\ &= f(3+3) \\ &= f(6) \\ &= ReLU(6) = 6 \end{aligned}$$

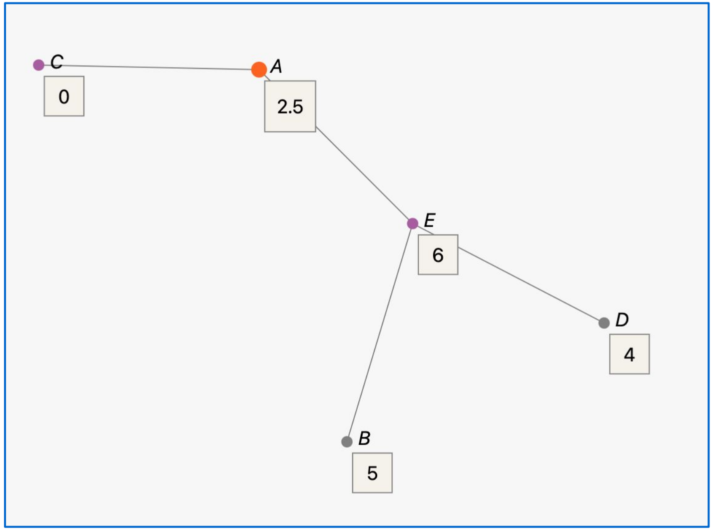
https://distill.pub/2021/understanding-gnns/

Graph Convolutions

Practical example on spatial GCN:



Iteration 0

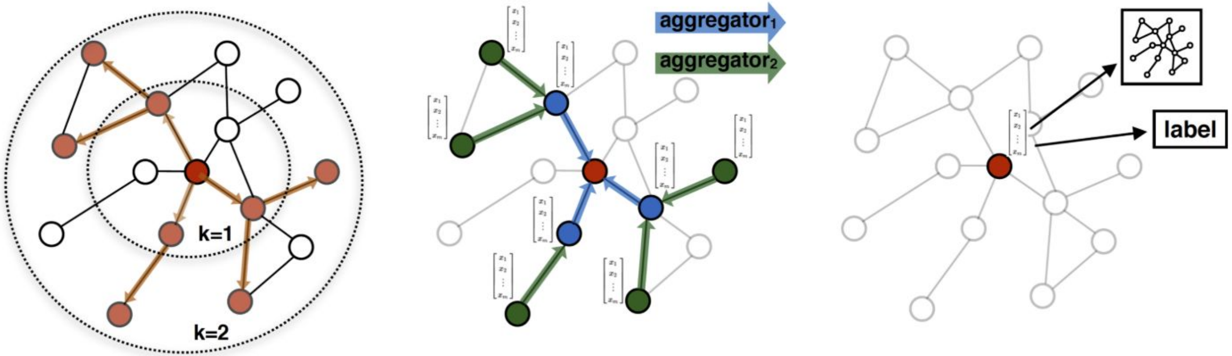


Iteration 1

https://distill.pub/2021/understanding-gnns/

Example 1: GraphSage

General idea:
Feature representation of one node gets updated by surrounding nodes



1. Sample neighborhood

2. Aggregate feature information from neighbors

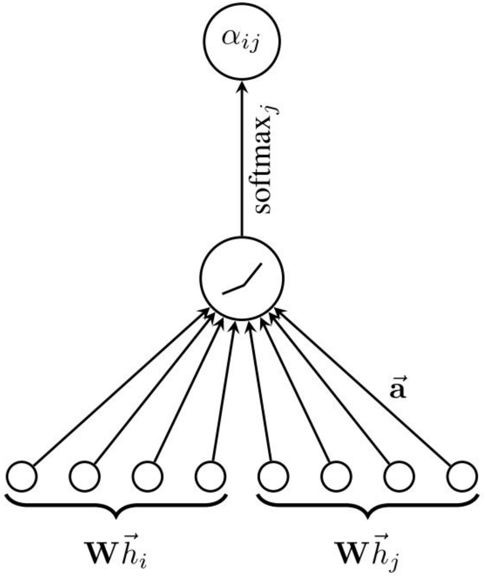
3. Predict graph context and label using aggregated information

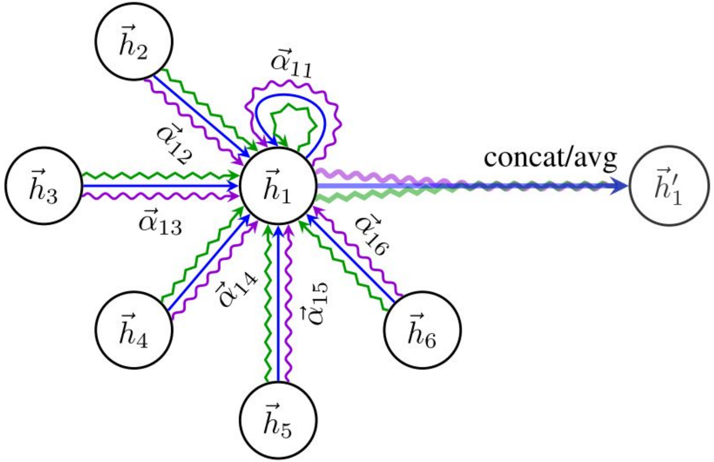
[1] W. L. Hamilton, R. Ying, J. Leskovec, Inductive Representation Learning on Large Graphs, NIPS 2017

Example 2: Graph Attention Network (GAT)

General idea:

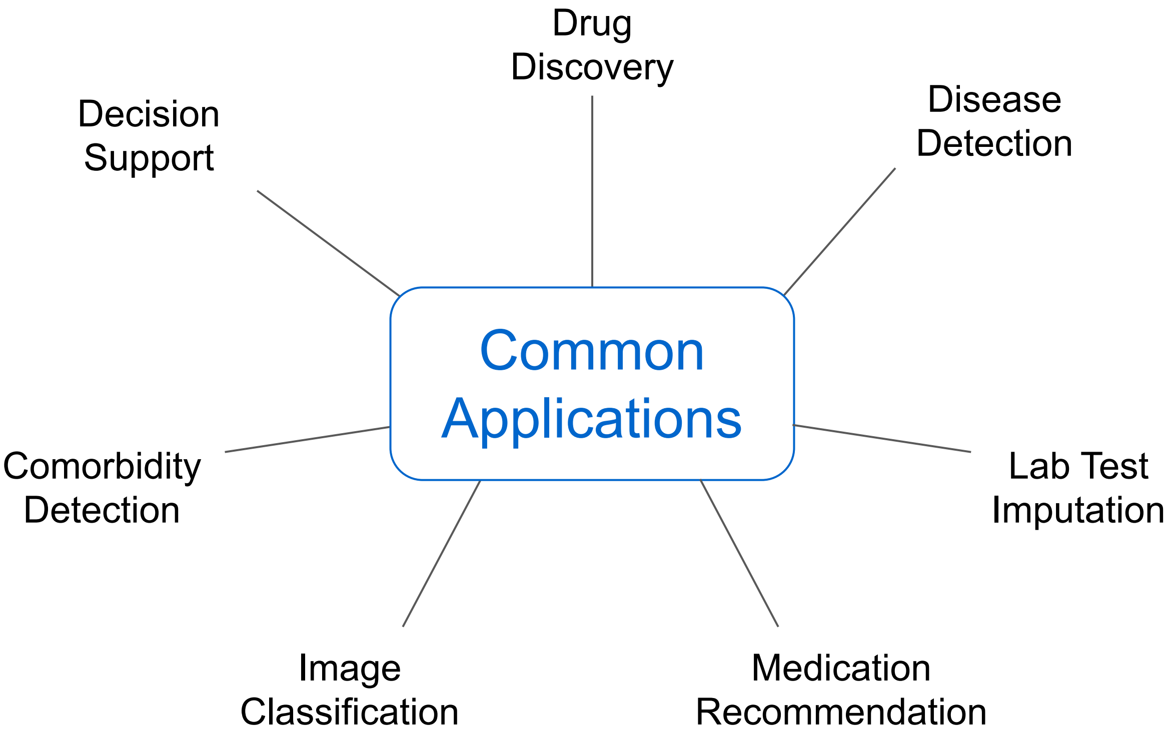
Network learns which neighbors are the most important for the update (attention





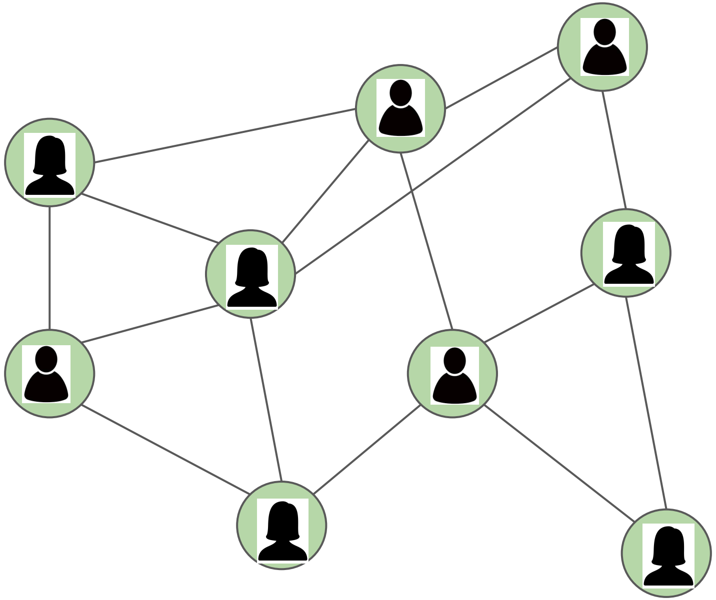
[1] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio and Y. Bengio, 2017. Graph attention networks ICLR 2018

Applications of GDL in Medicine



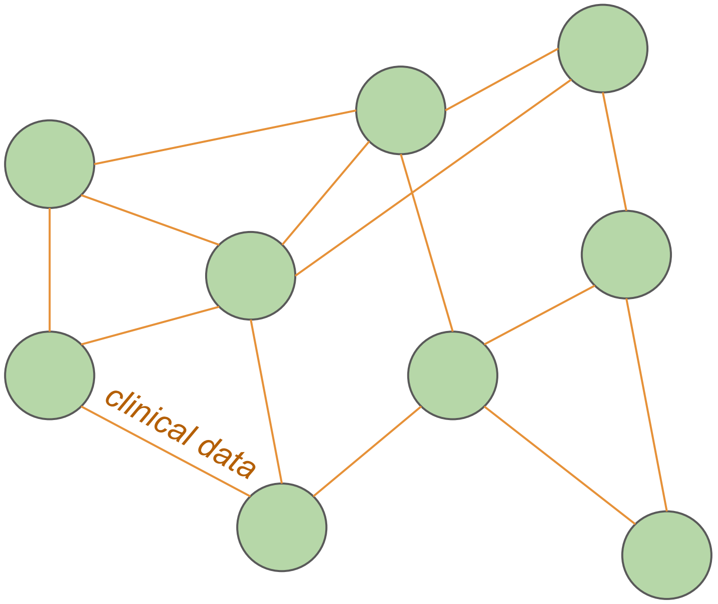
Population Graph

- Each subject is represented as one node in the graph
- Sharing information across whole patient population
- Combining multi-modal information,
  e.g. clinical data,
  imaging data,
  genetics,...



Population Graph - an Example

- Integration of clinical data and image data
- Subjects can be connected based on similarity e.g.



clinical data can be used to determine edges between similar subjects

Population Graph - an Example

- Integration of clinical data and image data
- Subjects can be connected based on similarity e.g.

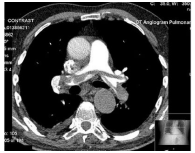
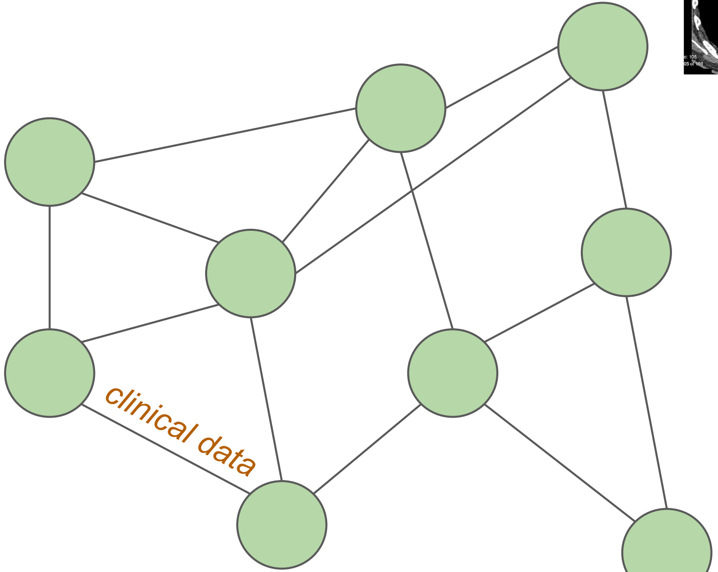
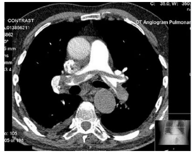


image information can be stored in node features



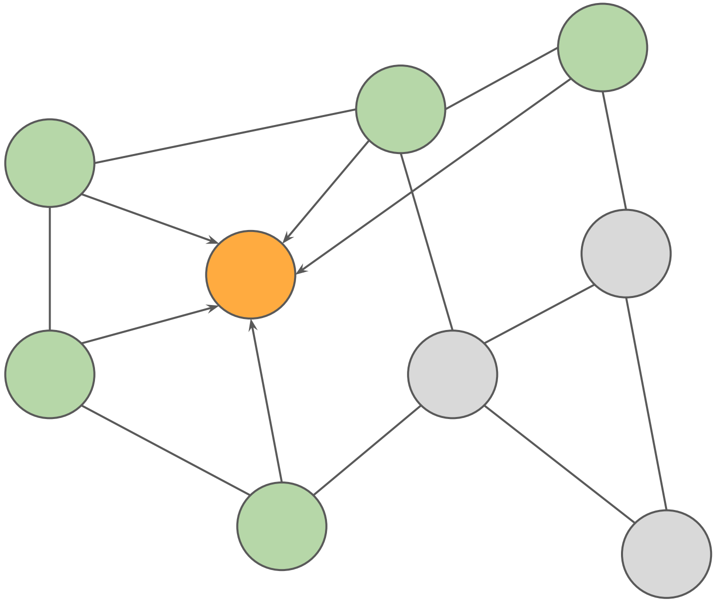


clinical data can be used to determine edges between similar subjects

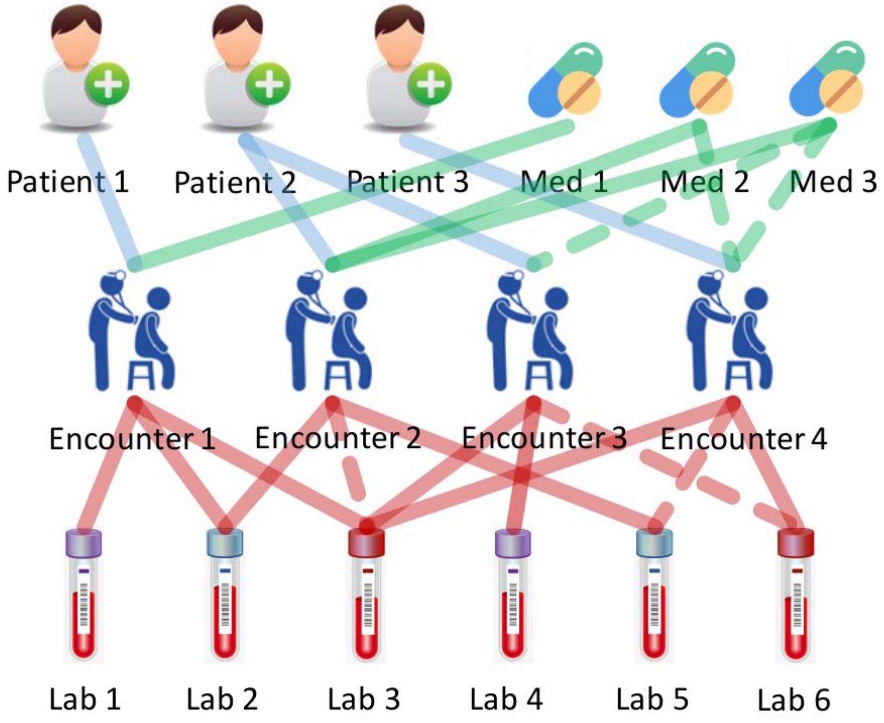
Population Graph - an Example

- Usage of **Graph Neural Networks (GNNs)** to learn patient specific predictions

- Idea: allow **information propagation** among neighbourhoods to learn from similar subjects

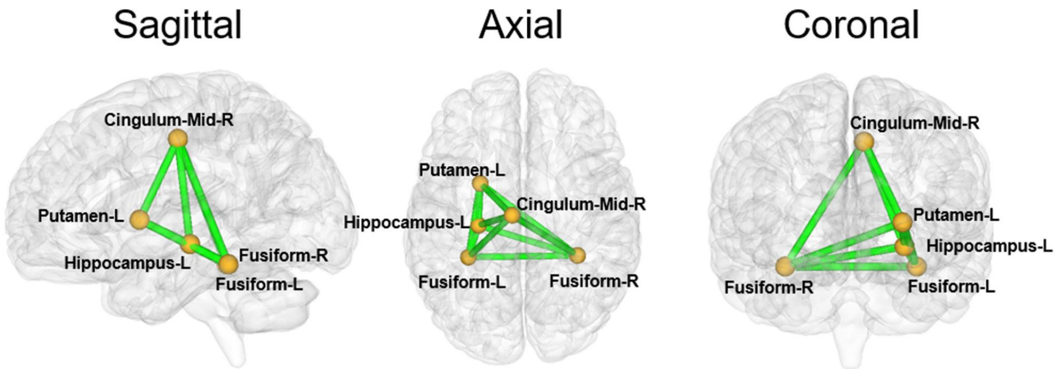


MedGCN: Medical Graph Convolutional Network



- Connecting **Patients**, **Encounters**, **Lab results** and **Medication**
- **Tasks**:
  - Medication recommendation
  - Lab test imputation

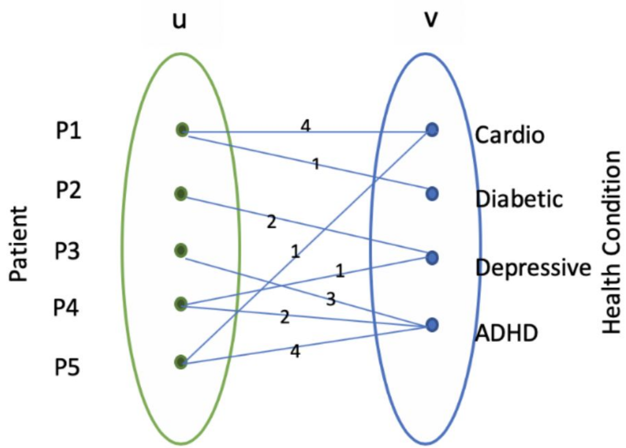
Disease Detection using Brain Connectivity Networks



- Detection of Discriminative Neurological Circuits
- Usage of fMRI Sequences
- Applications: Alzheimer’s Disease, Obsessive-Compulsive Disorder, Parkinson’s Disease, Autism Spectrum Disorder, etc.

Fig.: Xing, et al. "Detection of Discriminative Neurological Circuits Using Hierarchical Graph Convolutional Networks in fMRI Sequences.", 2020.,
Li, et al. "Graph neural network for interpreting task-fmri biomarkers.", 2019.,
Ma, et al. "Deep graph similarity learning for brain data analysis.", 2019.

Comorbidity Detection



a). Bipartite Graph G

Patient (u)	Cardio	Diabetic	Depressive	ADHD
P1	4	1	0	0
P2	2	0	1	0
P3	0	0	0	3
P4	0	0	1	2
P5	1	0	0	4

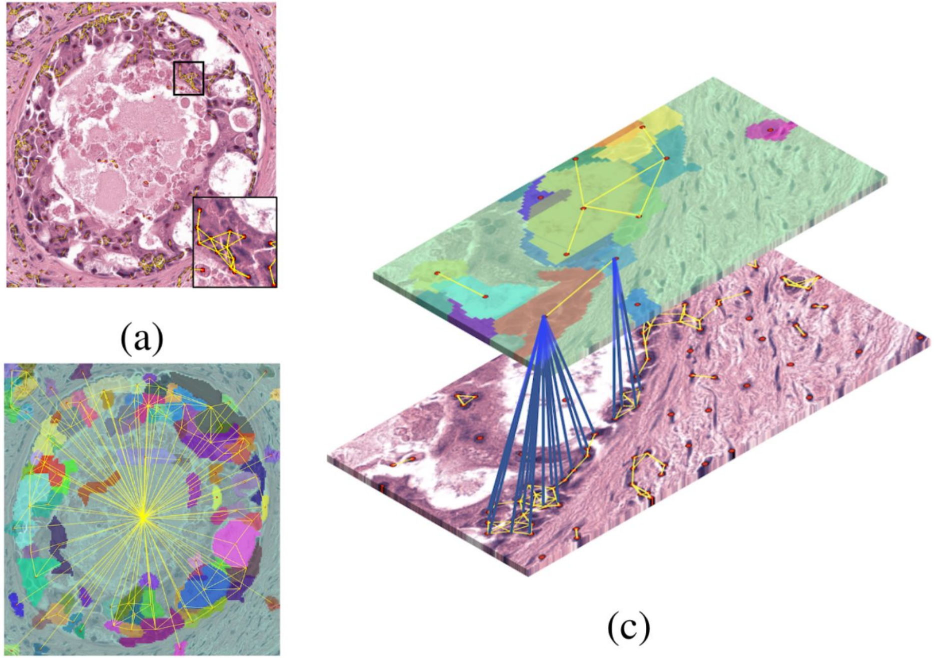
Health Condition or Adverse Event item (v)

b). Rating Matrix M

- Using matrix completion to predict unknown comorbidities
- Recovering missing links in the bipartite graph for health risk predictions

Wang, Rui, Ming-Ching Chang, and Marleen Radigan. "Modeling Latent Comorbidity for Health Risk Prediction Using Graph Convolutional Network.", 2020.

Histopathological Image Classification for Breast Cancer Subtyping



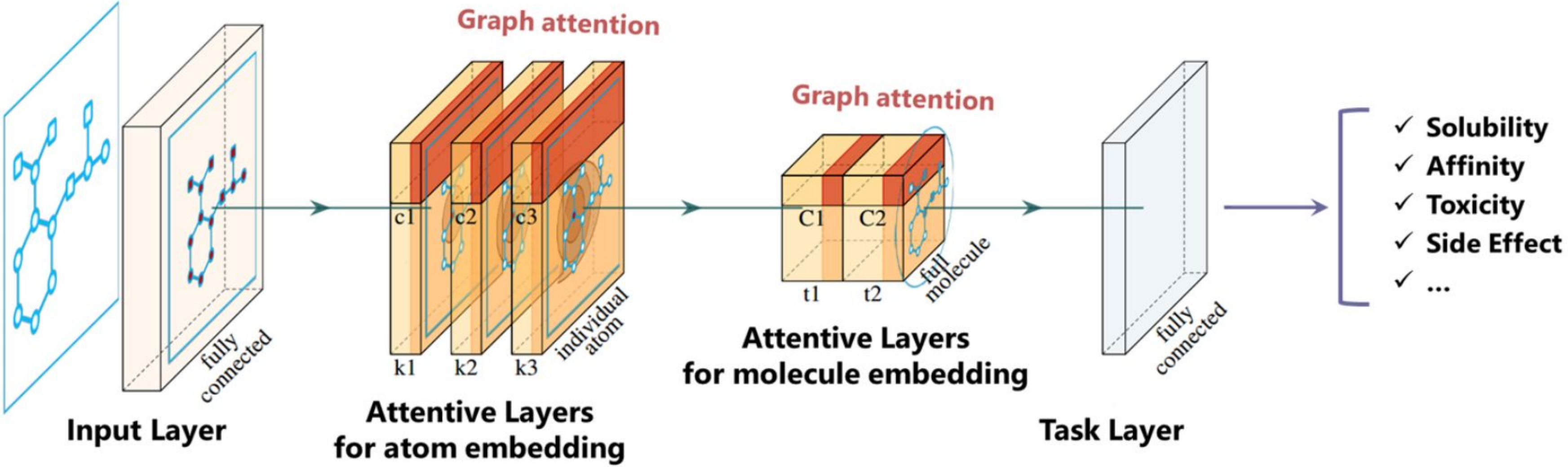
(a) (b) (c)

(a) **Cell Graph:** cells and cellular interactions

(b) **Tissue Graph:** region adjacency graph

(c) **HierArchical-Cell-to-Tissue** (HACT) **representation:** combination of (a) and (b)

Drug Discovery



- Using Graph Convolutional Networks to predict drug properties
- Capturing subtle substructure patterns

Xiong, et al. "Pushing the boundaries of molecular representation for drug discovery with the graph attention mechanism." 2019.

Recap:

Geometric Deep Learning

- allows to perform learning on datasets in non-Euclidean spaces
- operates on graphs, meshes, point clouds,...

Graph Neural Networks

- operate on graph-structured data specifically
- are used to solve three main tasks: node classification, graph classification, link prediction
- operate on spectral or spatial domain

Recap:

When?

- Intrinsic **non-Euclidean structure** of the dataset: molecules, social networks, meshes...
- **Multimodal applications**

How?

- Define **graph structure** if necessary: population graph, connectivity graph, region adjacency graph…
- Specify **node features** and **edges** (and potentially edge weights)
- Define the **type of problem**: node classification, graph classification, edge prediction
- Choose a graph learning **algorithm**
- **Train** the network :D

Pros and Cons

Pros

- Allows utilisation of complex inter-connected data
- Multimodal applications
- Expansion of Deep Learning to new fields → novel research
- Performance gain in some applications (e.g. social networks, molecule predictions)

Cons

- Sometimes the graph structure is not defined/unique
- Highly dependent on underlying structure of the data
- Fairly new → fewer content (tutorials, examples..) than other types of models.

Which application do we have?

Setting:

- We have clinical data
- and image data of 1000 patients

Goal:

- We want to predict the estimated time of survival for each patient

Which application do we have?

Setting:

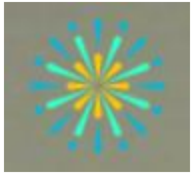
- We have clinical data
- and image data of 1000 patients

Goal:

- We want to predict the estimated time of survival for each patient

Node Classification

Edge Prediction



Graph Classification

Which application do we have?

Setting:

- We have clinical data
- and image data of 1000 patients

Goal:

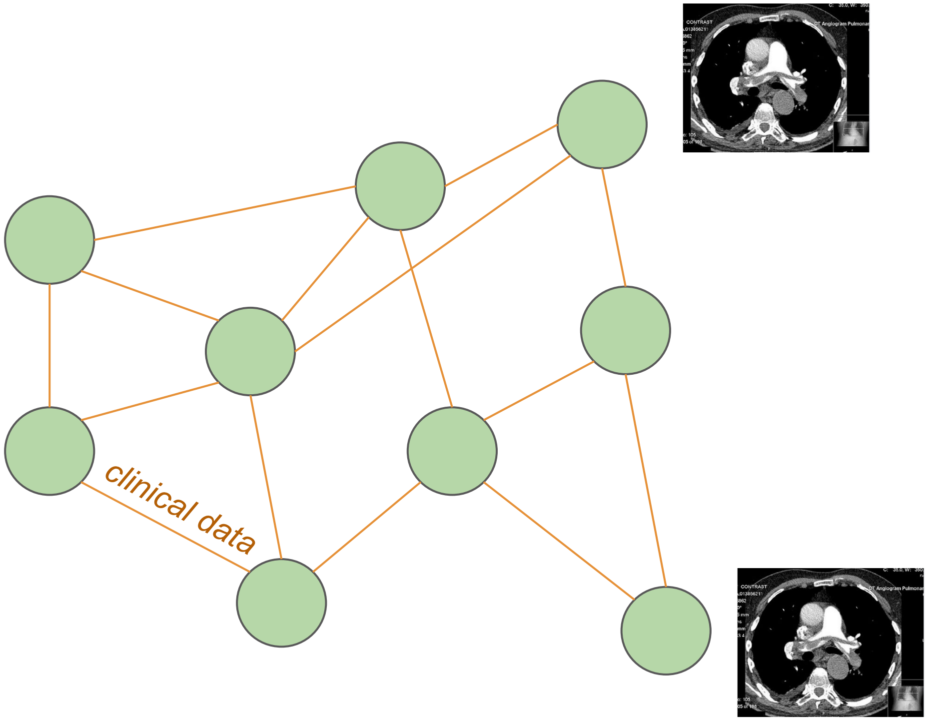
- We want to predict the estimated time of survival for each patient

Node Classification

Edge Prediction

Graph Classification

Patient Population Graph



Which application do we have?

Setting:

- We have fMRI data of brain scans of 100 patients
- We don't know any relationships between patients

Goal:

- We want to decide whether a patient has Alzheimer's Disease

Which application do we have?

Setting:

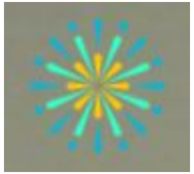
- We have fMRI data of brain scans of 100 patients
- We don’t know any relationships between patients

Goal:

- We want to decide whether a patient has Alzheimer’s Disease

Node Classification

Edge Prediction



Graph Classification

Which application do we have?

Setting:

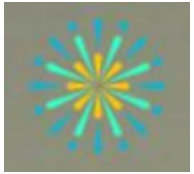
- We have fMRI data of brain scans of 100 patients
- We don't know any relationships between the patients

Goal:

- We want to decide whether a patient has Alzheimer's Disease

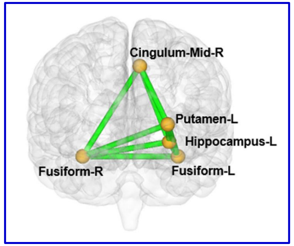
Node Classification

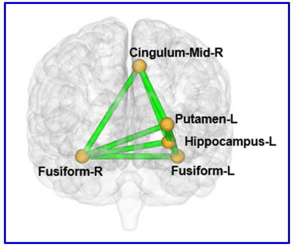
Edge Prediction

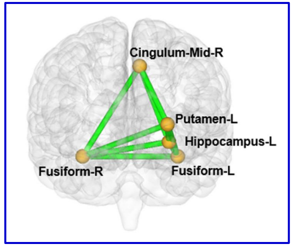


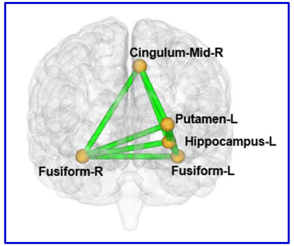
Graph Classification

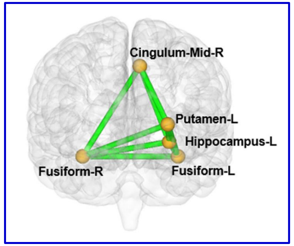
One Individual Graph per Patient











Alzheimer's? ✔ | Alzheimer's? ✘ | Alzheimer's? ✘ | Alzheimer's? ✔ | Alzheimer's? ✘

Future

Big picture thinking: Multimodal decision support in pathology

More validation: More population analyses

Covid Project: ICU admission, Mortality prediction

Cross modal retrieval: missing data

Current focus

EEG Signal Analysis: Epilepsy

Mahsa' work: analysis of GCNs

Interpretability with GCNs: Aiding to disease prediction

DG-GRU: Brain signals

Graph in Graph (ICL): Protein interactions

So far...

DGM: Population graph learning [Collaboration with ICL] — MICCAI 2020

AIG: Peri-diagnostic decision support — MICCAI 2020

ToxNet: Incorporating domain / textbook prior knowledge — MICCAI 2020

MC-SegGCN: Uncertainty based segmentation refinement — MIDL 2020

CNNGAT: Population regularized representation learning — MICCAI 2019

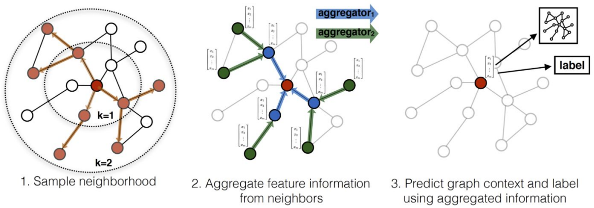
InceptionGCN: Handling graph heterogeneity — IPMI 2019

MultiGCN+PA: Personalized medicine, patient-specific attention — MICCAI 2019

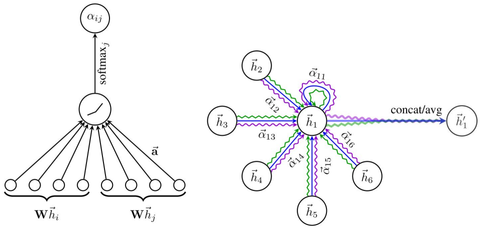
MultiGCN+SA: Increased robustness with multi-graphs and self-attention — ISBI 2019

GMC/MGMC: Handling missing data and imputation — GRAIL 2018 [J-BHI 2020]

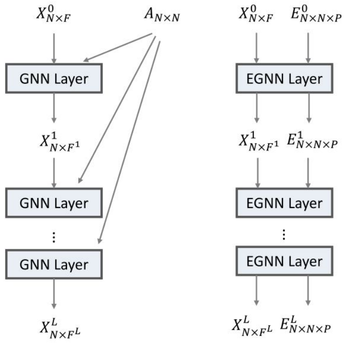
Advancements in the field of inductive graph learning



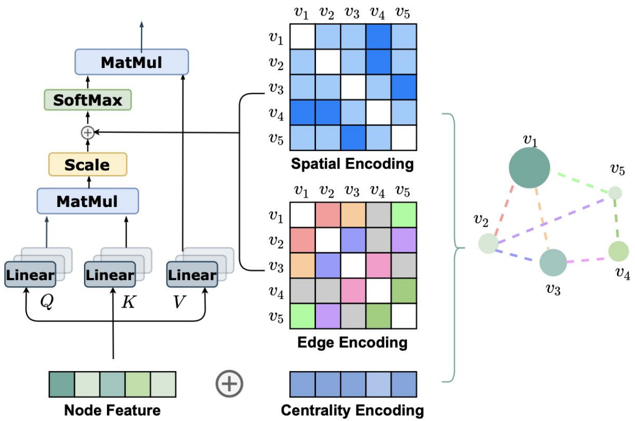
GraphSage, 2017 [1]



Graph Attention Networks, 2018 [2]



EGNN, 2019 [3]

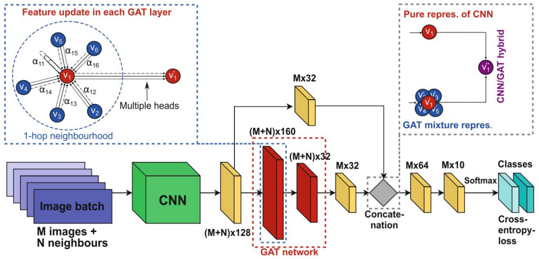


Graphormer, 2021 [4]

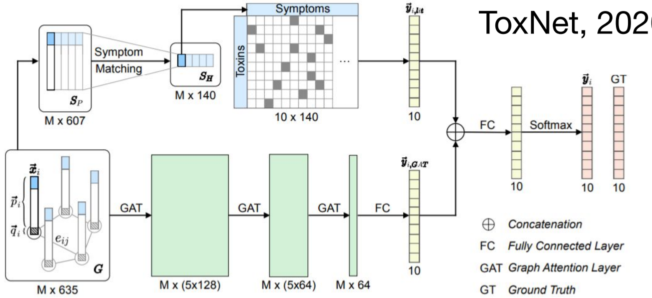
[1] W. L. Hamilton, R. Ying, J. Leskovec, Inductive Representation Learning on Large Graphs, NIPS 2017
[2] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio and Y. Bengio, 2017. Graph attention networks ICLR 2018
[3] Liyu Gong, Qiang Cheng, Exploiting Edge Features in Graph Neural Networks, CVPR 2019
[4] Ying, C., Cai, T., Luo, S., Zheng, S., Ke, G., He, D., ... & Liu, T. Y. (2021). Do Transformers Really Perform Badly for Graph Representation?. Advances in Neural Information Processing Systems, 34.

Advancements in the field of inductive graph learning

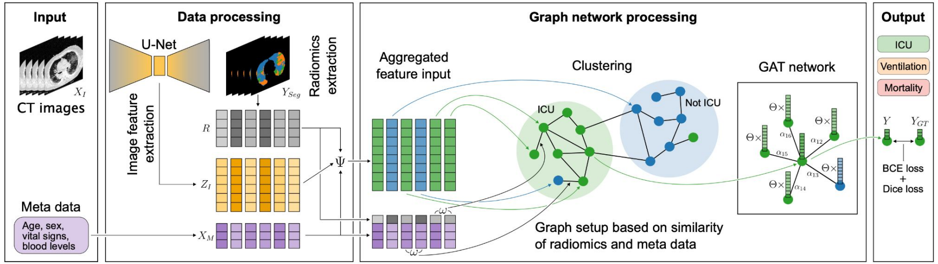
CNNGAT, 2019 [1]



ToxNet, 2020 [2]

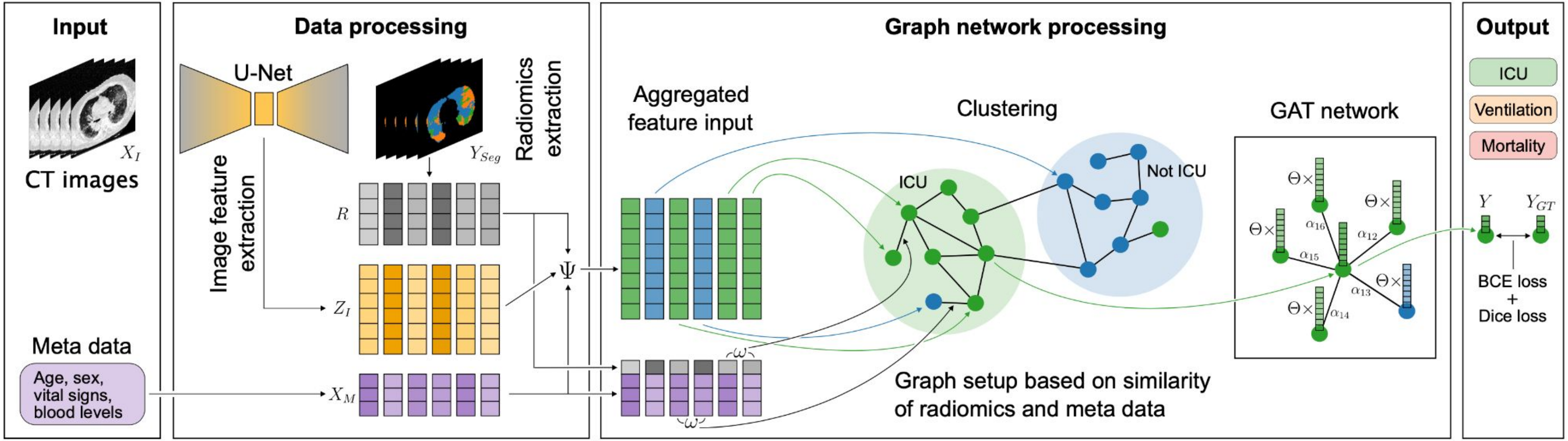


U-GAT, 2021 [3]



[1] H. Burwinkel, A. Kazi, G. Vivar, S. Albarqouni, G. Zahnd, N. Navab, S. A. Ahmadi, Adaptive Image-Feature Learning for Disease Classification Using Inductive Graph Networks, MICCAI 2019
[2] H. Burwinkel, M. Keicher, D. Bani-Harouni, T. Zellner, F. Eyer, N. Navab, S. A. Ahmadi, Decision Support for Intoxication Prediction Using Graph Convolutional Networks, MICCAI 2020
[3] Keicher, M., Burwinkel, H., Bani-Harouni, D., Paschali, M., Czempiel, T., Burian, E., ... & Wendler, T. (2021). U-GAT: Multimodal Graph Attention Network for COVID-19 Outcome Prediction. arXiv preprint arXiv:2108.00860.

U-GAT: Multimodal Graph Attention Network for COVID-19 Outcome Prediction

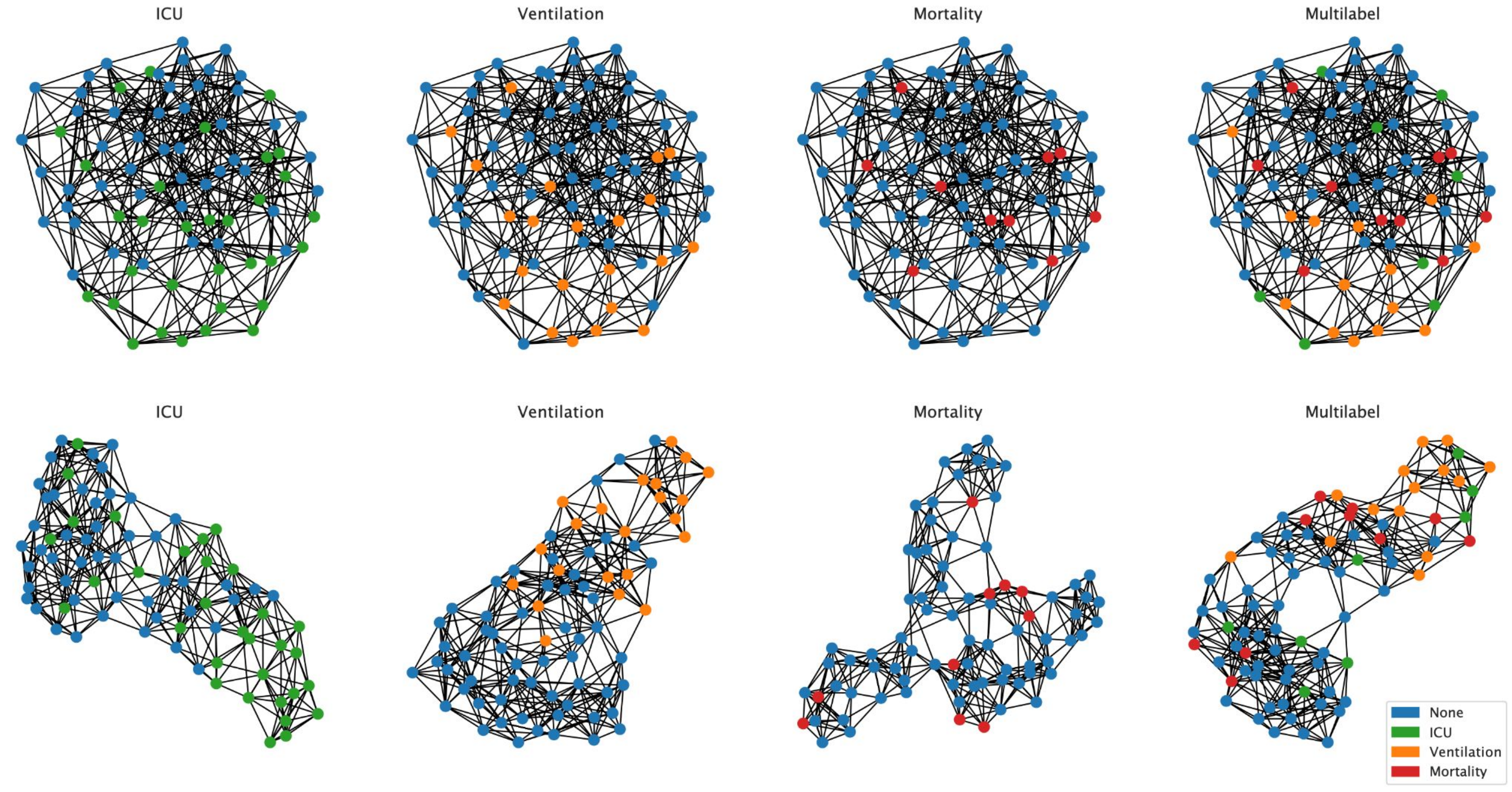


Keicher, M., Burwinkel, H., Bani-Harouni, D., Paschali, M., Czempiel, T., Burian, E., ... & Wendler, T. (2021). U-GAT: Multimodal Graph Attention Network for COVID-19 Outcome Prediction. *arXiv preprint arXiv:2108.00860*.

KNN Graph Construction with Mutual Information Weighted Features

KNN

MI weighted KNN

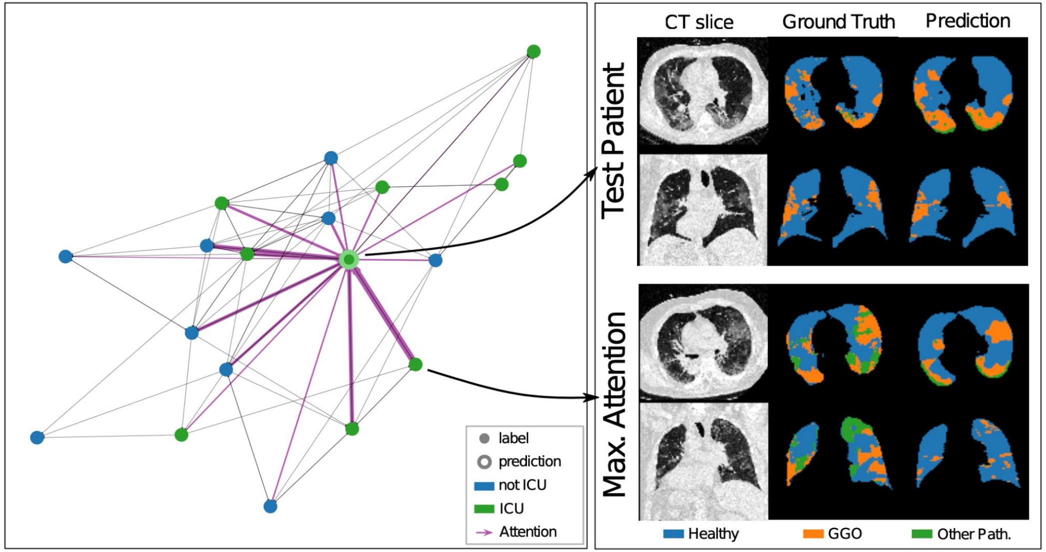


Keicher, M., Burwinkel, H., Bani-Harouni, D., Paschali, M., Czempiel, T., Burian, E., ... & Wendler, T. (2021). U-GAT: Multimodal Graph Attention Network for COVID-19 Outcome Prediction. *arXiv preprint arXiv:2108.00860*.

Results ICU Prediction

Task	Architecture	AP	DICE
Segmentation	U-Net	-	**64.14 ± 1.76**
ICU	MLP	57.65 ± 10.84	–
ICU	ResNet18	66.97 ± 9.68	–
ICU	U-Net*+KNN	61.17 ± 9.94	(64.14 ± 1.76)
ICU	U-Net*+MLP	61.50 ± 12.69	(64.14 ± 1.76)
ICU	U-Net*+GCN	68.75 ± 15.07	(64.14 ± 1.76)
ICU	ResNet18-GAT	63.70 ± 16.52	–
ICU	U-Net*+GAT	69.80 ± 12.04	(64.14 ± 1.76)
ICU+Seg.	U-Net-GAT	**69.94 ± 14.85**	61.39 ±2.16
Multilabel+Seg.	U-Net-GAT	64.89 ± 12.82	60.70 ± 1.85

* end-to-end training

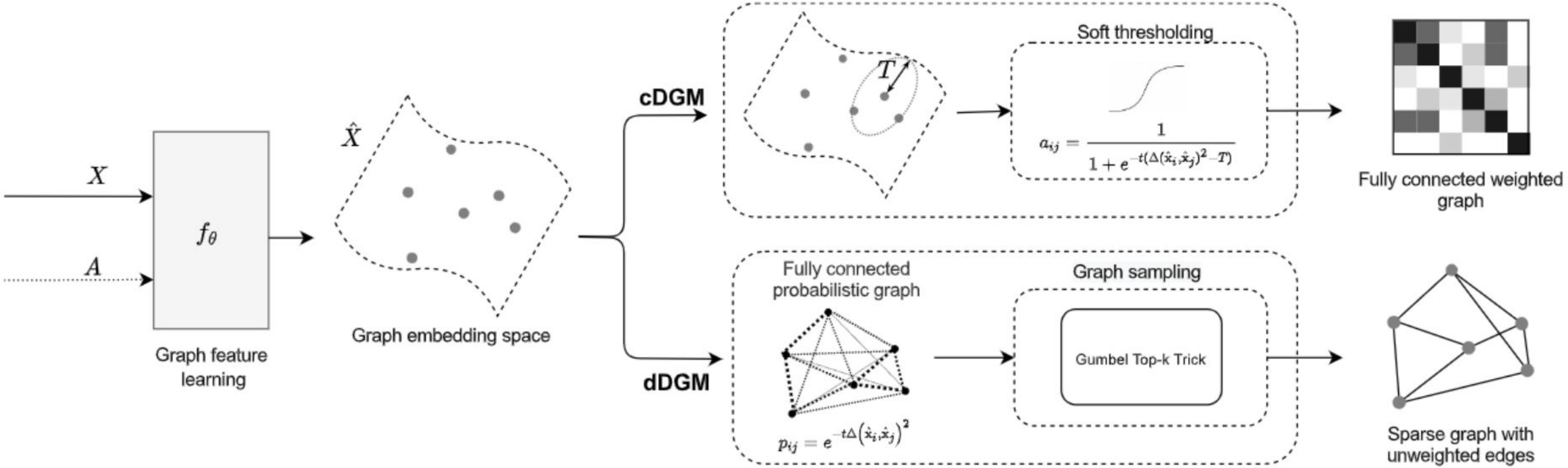


	ICU	Healthy (Prediction)	GGO (Prediction)	Other Path. (Prediction)	IL-6	CRP	Attention
Test Patient	yes	81.1% (76.9%)	16.9% (20.5%)	2.0% (2.3%)	89.0	16.2	0.159
Max. Attention	yes	49.7% (74.5%)	25.3% (19.1%)	24.9% (6.4%)	135.0	13.3	0.134

Fig. 5. Left: Batch graph showing the attention scores of a single test patient. The thickness of the line corresponds to the attention score of the respective neighbors after two hops. Right: CT images, segmentation ground truth and predicted segmentation of a single axial and coronal slice from the test patient and the neighbor with maximum attention. Bottom: Most important features for the test patient and the neighbor with maximum attention. In brackets, the radiomics predicted by the pretrained U-Net are shown.

Keicher, M., Burwinkel, H., Bani-Harouni, D., Paschali, M., Czempiel, T., Burian, E., ... & Wendler, T. (2021). U-GAT: Multimodal Graph Attention Network for COVID-19 Outcome Prediction. *arXiv preprint arXiv:2108.00860*.

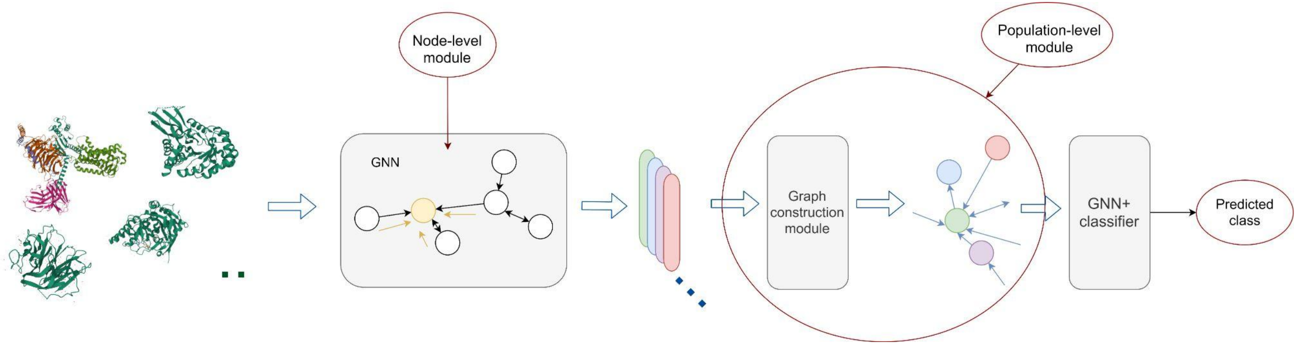
Differentiable graph module (DGM) for graph convolutional networks.



Benchmark datasets: CiteSeer, PubMed, Cora.
Medical Datasets: Tadpole, UK Biobank.
Computer Vision Datasets: Shapenet, Animals with Attribute 2 dataset

Kazi, A., Cosmo, L., Navab, N. and Bronstein, M., 2020. Differentiable graph module (dgm) for graph convolutional networks. *arXiv preprint arXiv:2002.04999*. Accepted in PAMI 2022.

Graph-in-graph



- HCP +4.1
- Tox21 +4.3
- PROTEINS +2.1
- NCI1 +1.8
- DD +0.1

Main idea is to obtain the population graph on graph-based data by learning the graph structure and ensuring normal nodes degree distribution.

Mullakaeva, K., Cosmo, L., Kazi, A., Ahmadi, S. A., Navab, N., & Bronstein, M. M. (2022). Graph-in-Graph (GiG): Learning interpretable latent graphs in non-Euclidean domain for biological and healthcare applications. *arXiv preprint arXiv:2204.00323*.

Time for Questions