Building & Mining Knowledge Graphs

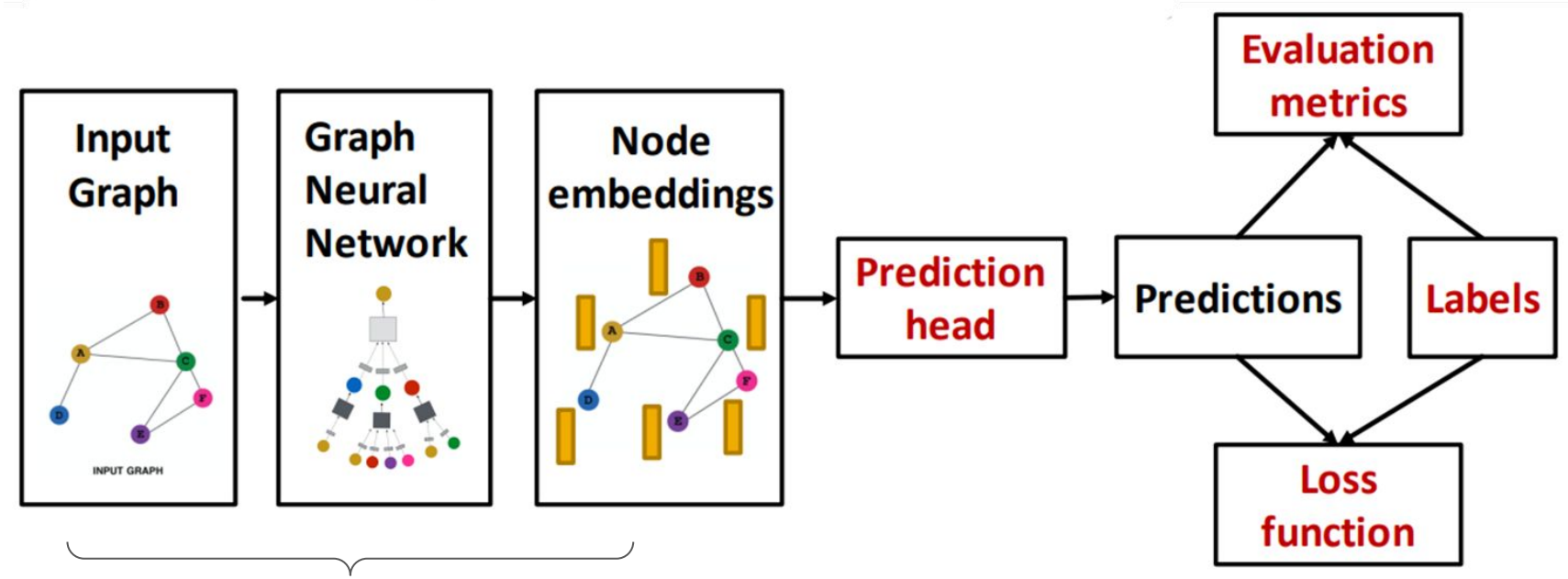
(KEN4256)

Lecture 8: Link Prediction & Explainable AI

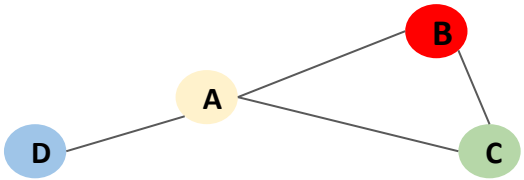
Maastricht University
Institute of Data Science

id: KEN4256_L8
version: 1.2024.0
created: February 9, 2019
last modified: March 26, 2024
published on: March 26, 2024

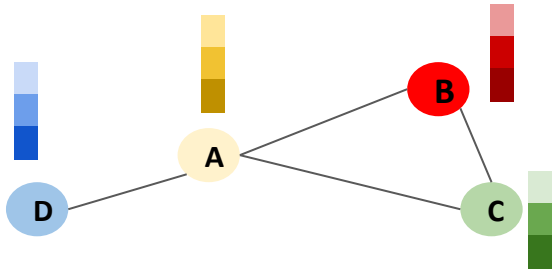
Training Framework



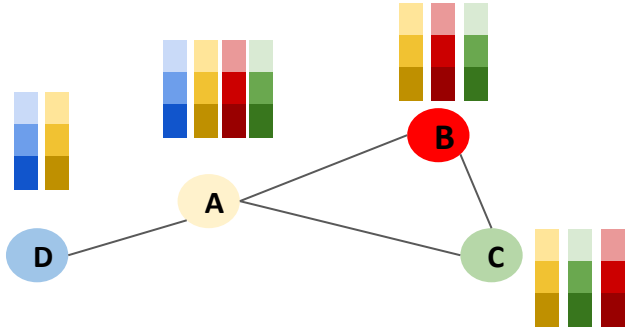
Forward Pass



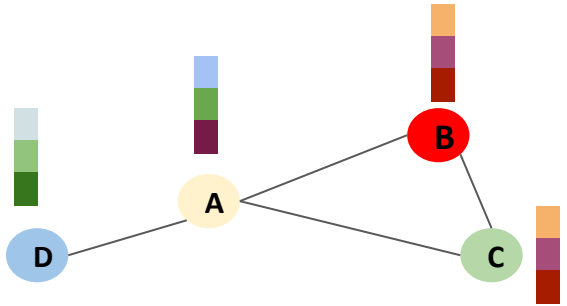
Compute messages



Propagate messages

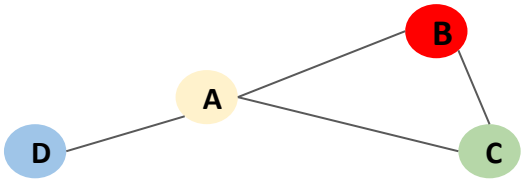


Aggregate

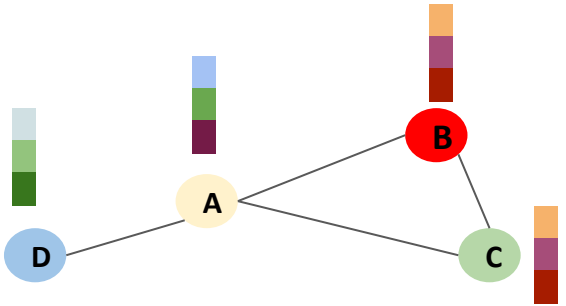


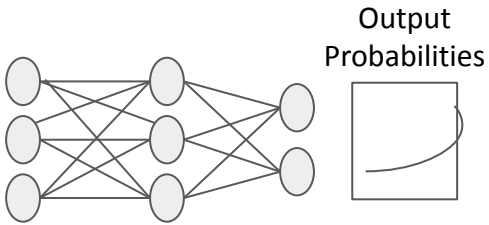
Prediction & Learning

The goal of learning is to minimize the loss between predictions and labels:



Forward Pass





Prediction Head is typically a fully connected layer (of size embedding dimension x number of classes), followed by prediction function

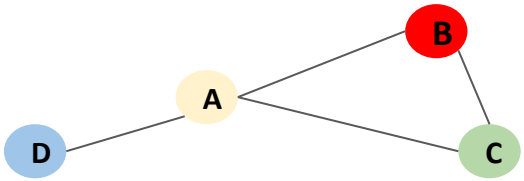
Prediction Head & Loss Computation

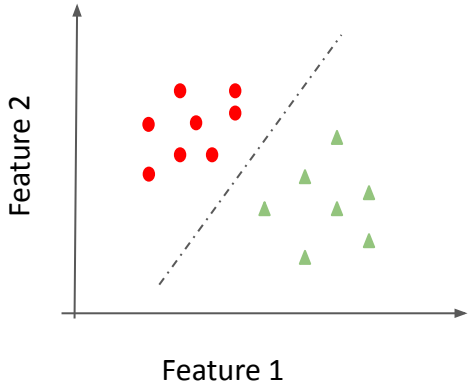
Node	Embeddings	Output	Labels
A		0.4	1
B		0.7	1
C		0.91	1
D		0.1	0

Backpropogate these errors to update the parameters

2D Interpretation

With linear layers, this process is the same as learning embeddings such that node representations are linearly separable by a hyper-plane, i.e. **binary/multi-label classification with graph input**





Machine Learning GNN-related Tasks

Node Classification - Predict a property of a node ✅

e.g. categorize online users/items

Link Prediction - Predict whether there are missing links between two nodes

e.g. knowledge graph completion

Graph Prediction - Categorize different graphs

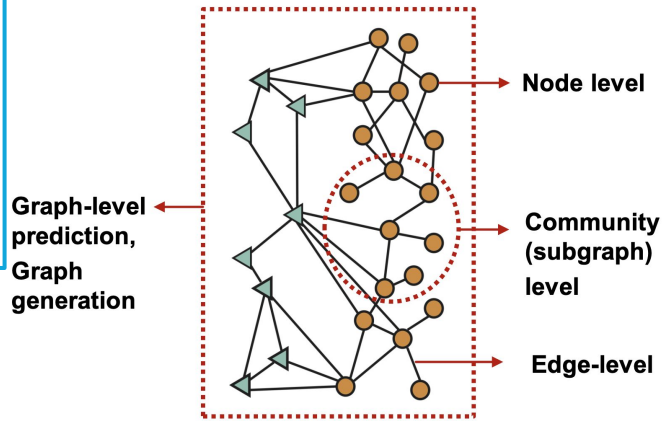
e.g. Protein function prediction

Clustering - Detects if nodes are from a community

e.g. social circle detection

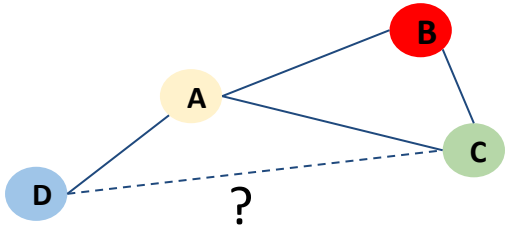
Graph Generation - Create new graphs with shared structure

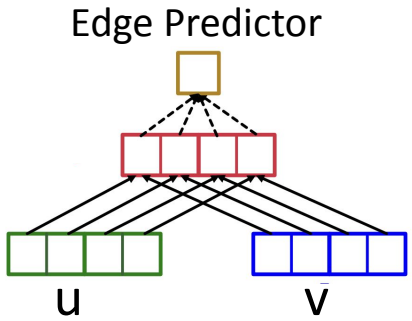
e.g. drug discovery



Link Prediction

- If there is only one edge type, the **goal** is to predict missing edges
- Split edges into train/validation/test sets
  - In each group use certain edges as always fixed and the rest for supervision
- The same Encoder and Decoder architecture as before applied
- **Decoder** output predicts the probability for existing edge
  - Alternatively, use Concat(u,v) passed through fully connected layer and sigmoid function

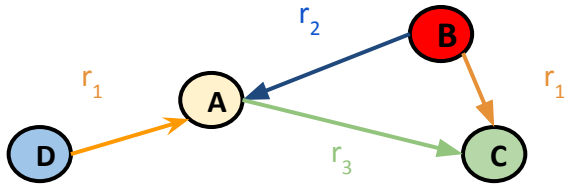




Heterogeneous Graphs

- How to handle graphs with multiple types?
- Captures different types of interactions between entities
- Knowledge Graphs are HeteroGraphs

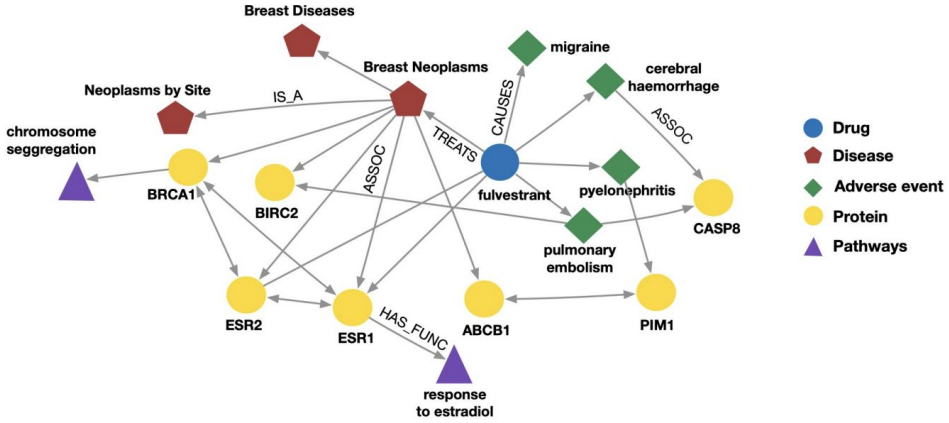
e.g Google Knowledge Graph , Amazon Product Graph, Facebook Graph API , IBM Watson, Project Hanover/Literome, Project Hanover/Literome

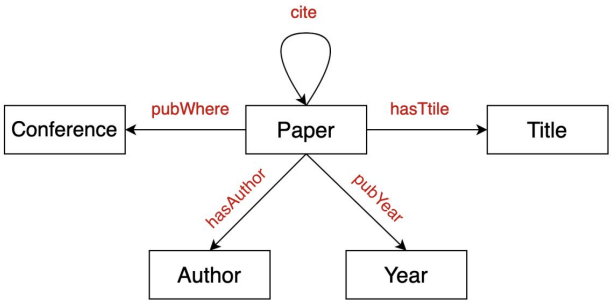


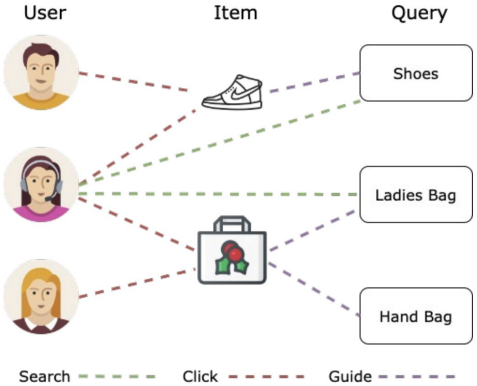
amazon Google facebook LinkedIn

Heterogeneous Graphs

- Many real life applications, e.g. Biomedical Knowledge Graphs, Events Graphs, Academic Graphs
- More expensive (computation, storage), more complex implementation

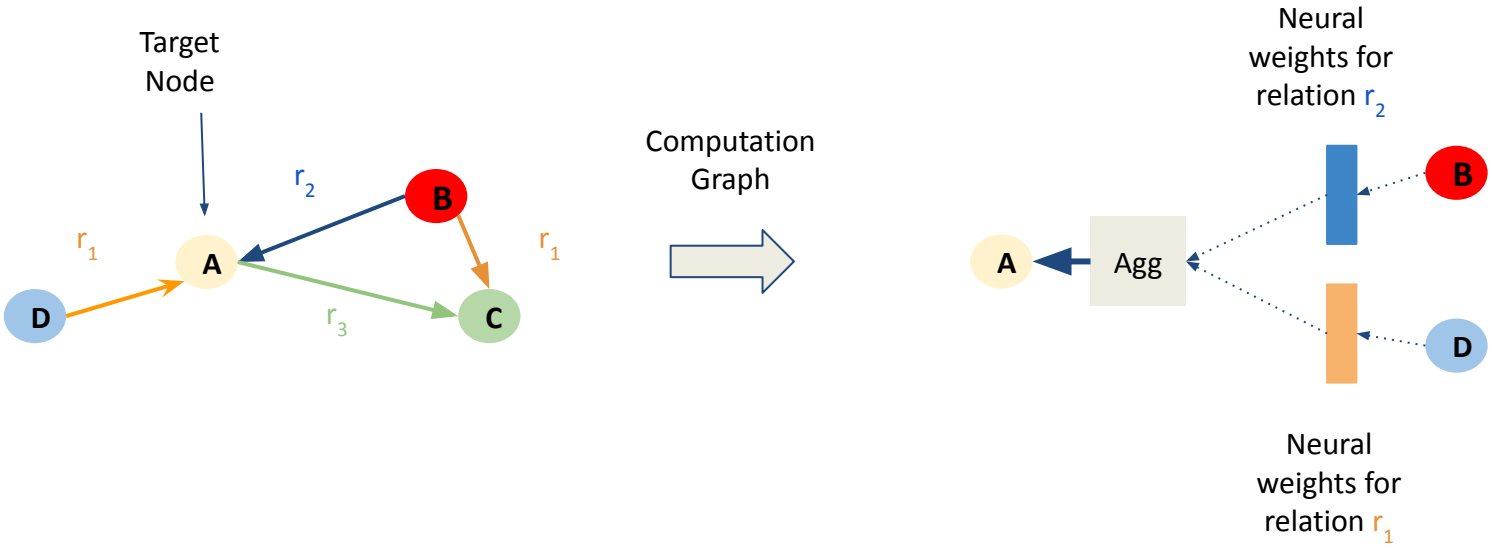






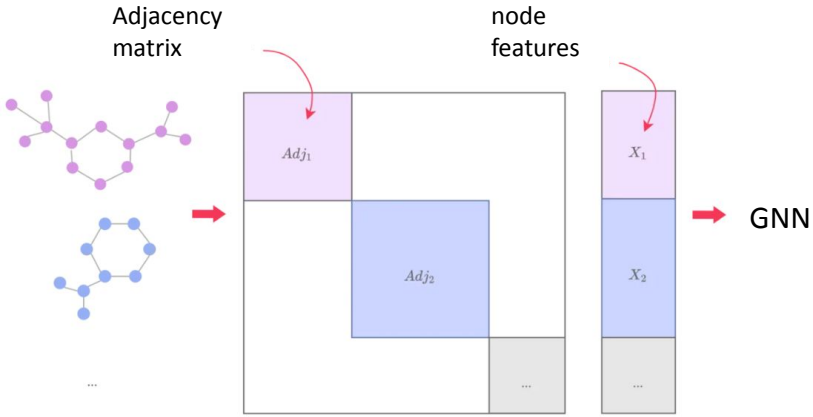
Relational GCN

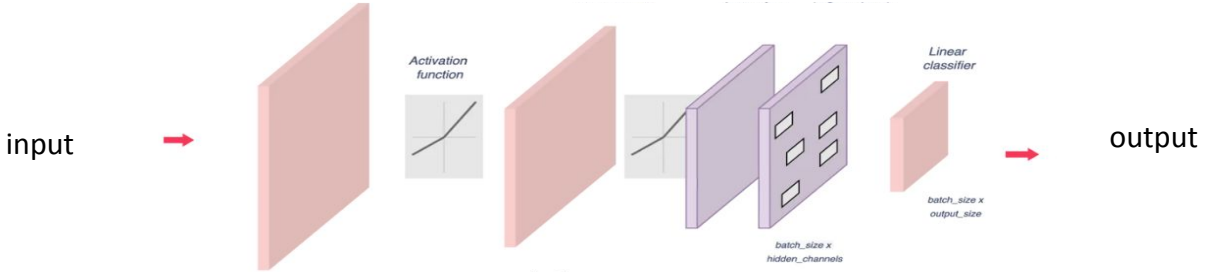
- Extension of Graph Convolutional Networks to handle heterogeneous graphs with multiple edge and relation types
- **Key Idea**: Use different neural network weights for different relation types



Graph-level Prediction

- To perform graph-level prediction, we need to <u>learn from many graphs</u>
- Same GNN methodology can be utilized with the following modifications
  - Stack adjacent matrices in a diagonal manner leading to a large graph with isolated subgraphs
  - Concatenate node features

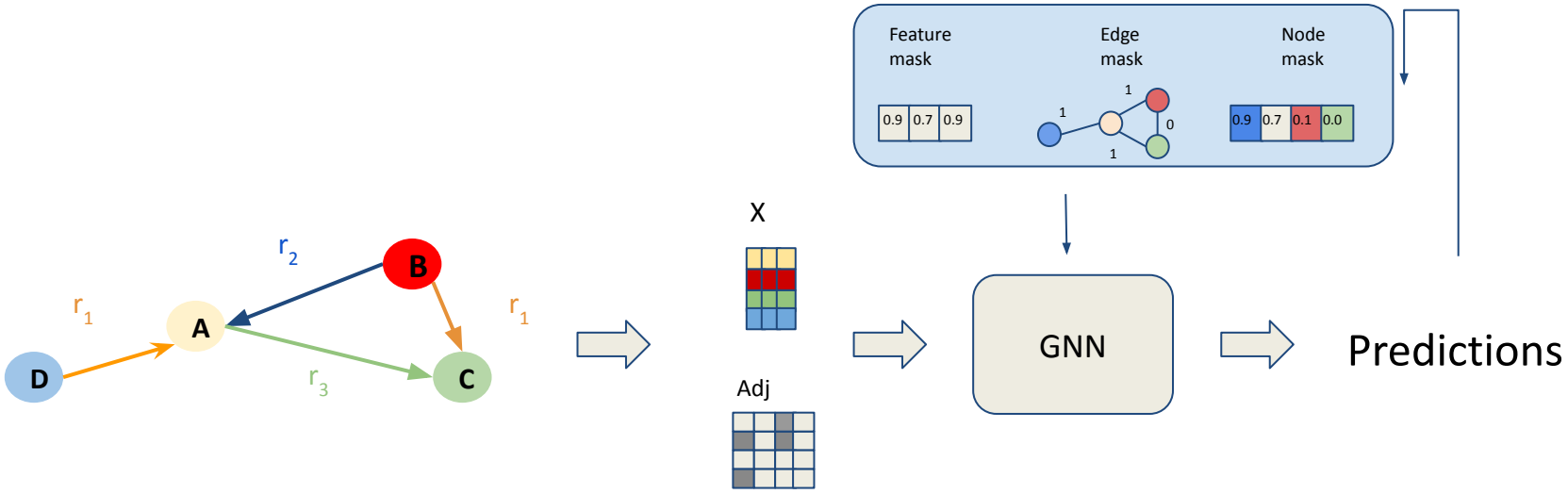




https://blog.dataiku.com/graph-neural-networks-part-three

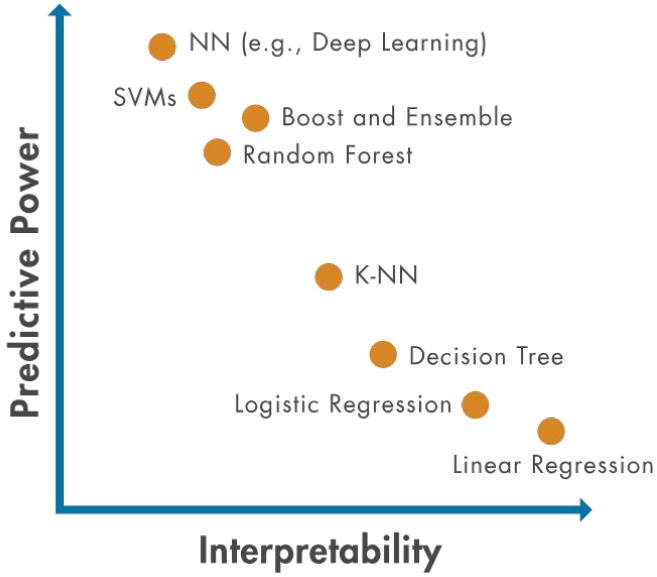
Explainable AI

- Explain how model makes prediction, without necessarily understanding the inner mechanics
- Model-agnostic (need only outputs) vs model specific



Explainable AI

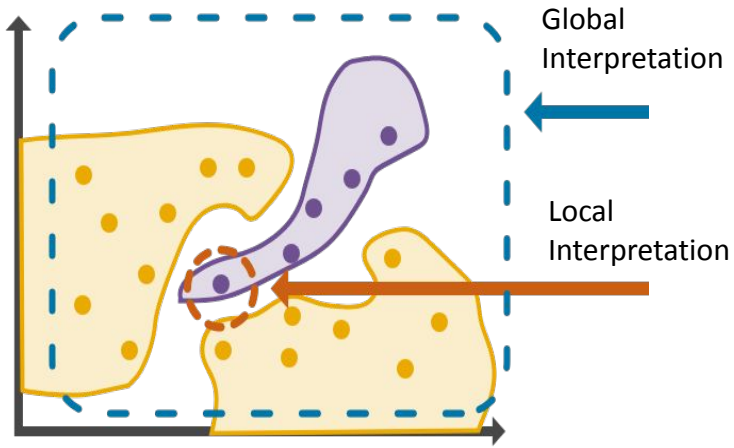
- Explain how model makes prediction, without necessarily understanding the inner mechanics
- Model-agnostic (need only outputs) vs model specific
- Some models are interpretable by nature
  - e.g. decision tree



https://nl.mathworks.com/discovery/interpretability.html

Explainable AI

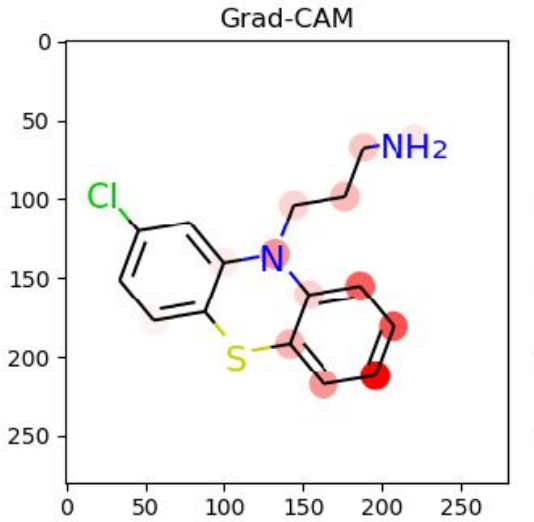
- Explain how model makes prediction, without necessarily understanding the inner mechanics
- Some models are interpretable by nature
  - e.g. decision tree
- **Instance** (local) vs **Model** (global)
  - Explanation at the level of an individual prediction or the most influential parameters from the whole model



GNN Explanations

Instance Level

- **Gradient based,** rely on gradients to determine input importance , e.g. CAM, Grad-CAM, Buided BP

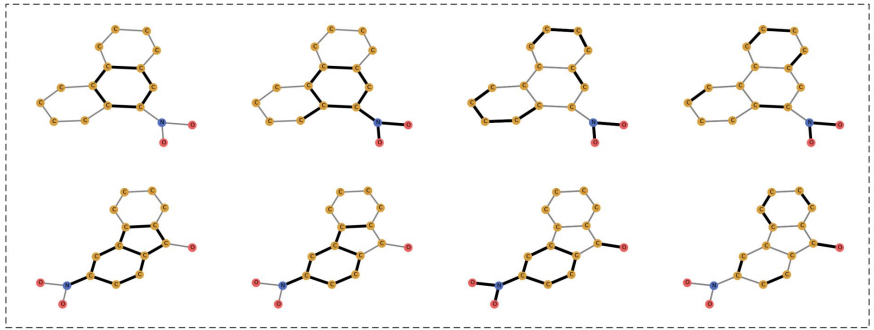


https://github.com/ndey96/GCNN-Explainability

GNN Explanations

Instance Level

- **Gradient based,** rely on gradients to determine input importance , e.g. CAM, Grad-CAM, Buided BP
- **Feature based,** rely on features and interpolation to determine input importance, e.g. GraphMask, SubgraphX, GNNExplainer

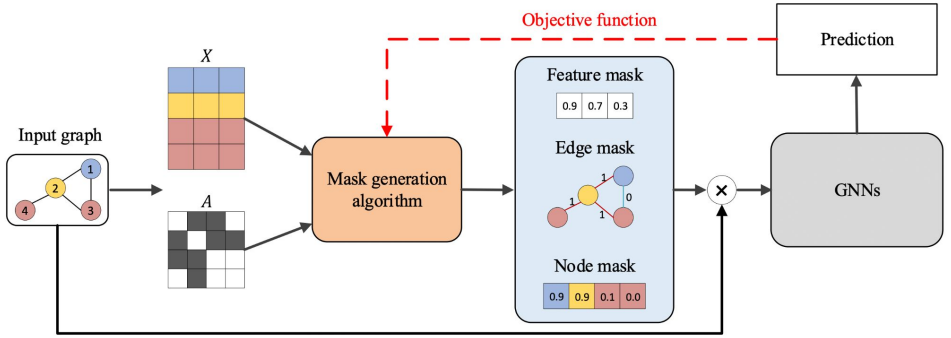


https://arxiv.org/pdf/2102.05152.pdf

GNN Explanations

Instance Level

- **Gradient based,** rely on gradients to determine input importance , e.g. CAM, Grad-CAM, Buided BP
- **Feature based,** rely on features and interpolation to determine input importance, e.g. GraphMask, SubgraphX, GNNExplainer
- **Perturbation based,** examine output variations w.r.t. Input perturbations

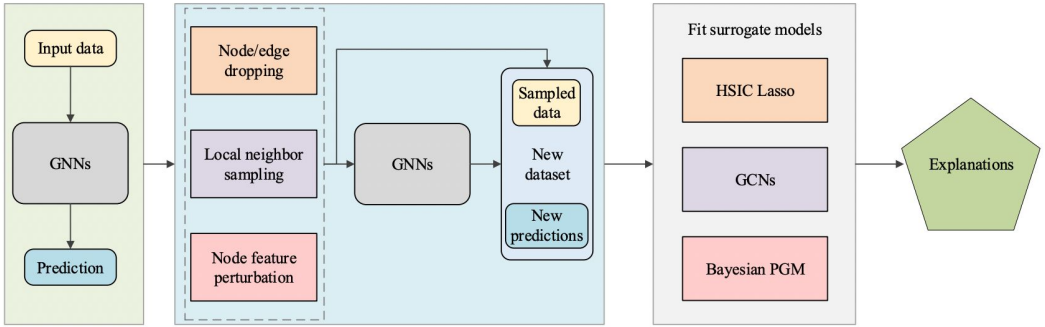


Yan et al, "Explainability in GNN: A Taxonomic Survey"

GNN Explanations

Instance Level

- **Gradient based,** rely on gradients to determine input importance , e.g. CAM, Grad-CAM, Buided BP
- **Feature based,** rely on features and interpolation to determine input importance, e.g. GraphMask, SubgraphX, GNNExplainer
- **Perturbation based,** examine output variations w.r.t. Input perturbations
- **Surrogate Model,** train a simpler, interpretable model to approximate predictions, e.g. GraphLIME



Yan et al, "Explainability in GNN: A Taxonomic Survey"

Saliency Map

- Saliency Maps originates from computer vision-related literature and is a **pixel attribution method**, which highlights the pixels that were relevant for a certain image classification by a neural network
- The idea is to compute the gradients of the output with respect to the input
- Then, the attribution value along the ith dimension for an input $x \in R^n$ is defined as the absolute value of the gradient:

$$\text{Saliency}_i(x) = \left|\frac{\partial F(x)}{\partial x_i}\right|, \text{ where } F(x) \text{ denotes the output of the GNN model on input } x.$$

Integrated Gradients

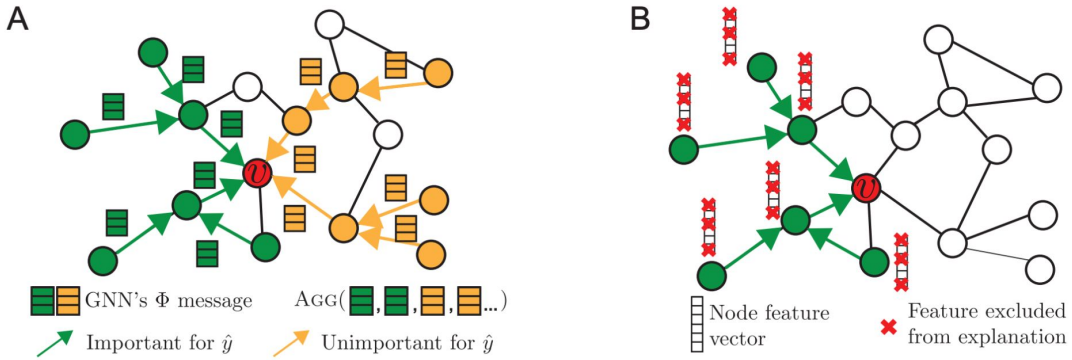
- Saliency Maps are not well localized, no guarantees, and unstable w.r.t. Small changes and sensitivity in input
- Extension of Saliency Maps, by integrating along a path from an all masked (empty) graph to the target graph
- In essence, **average saliency maps** over many graph topographies

$$G_A = (G - G') \int_{\alpha=0}^{1} \frac{dy(G' + \alpha(G - G'))}{dG} d\alpha$$

Leneling et al, “Evaluating Attributions for GNNs”

GNNExplainer

- If the graphs are small, saliency maps work well
- GNNExplainer, treats the problem of finding masks as an optimization problem, subject to following regularizations:
  - Masks should not change original predictions
  - Masks should be as small and sparse as possible

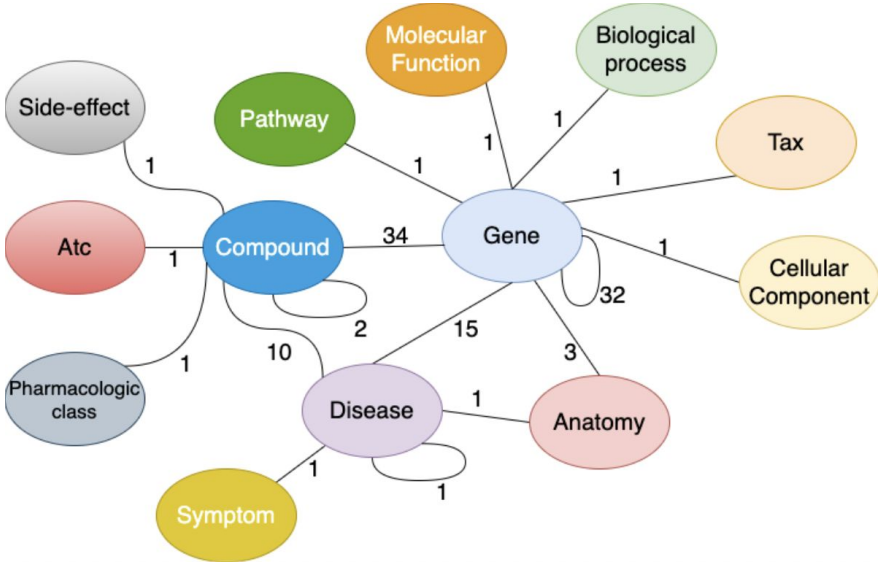


Ying et al, "GNN Explainer: Generating Explanations for GNNs"

Another Example of Heterogeneous Graph

- Drug Repurposing aims to find new potential uses for an existing drug and is often modeled as **a link prediction problem** where the link between the candidate drug and the target disease is aimed to be predicted:

  (Compound, treats, Disease)



Interconnections among entities in DRKG.

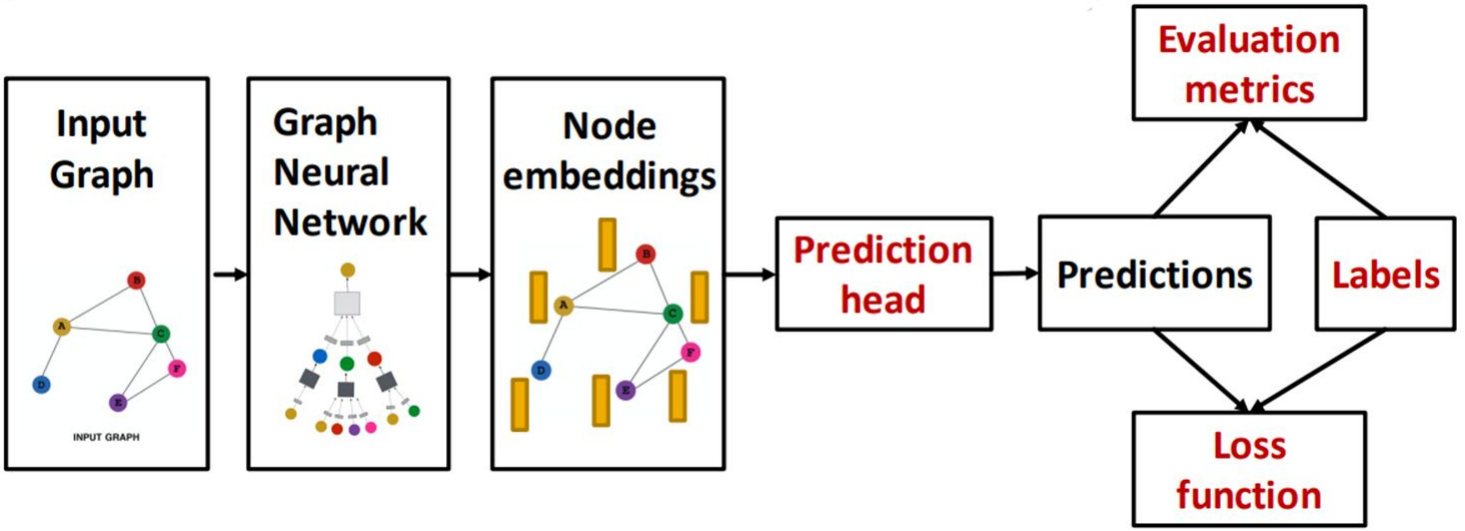
Loech J, Yang Y, Ekmei P, Dumontier M, and Celebi R, “Explaining Graph Neural Network Predictions for Drug Repurposing”, SWAT4HCLS 2024

Methodology

- Use **Graph Neural Networks** to capture **semantics**, **graph structure** and **relationships between nodes**
- Apply **Saliency Maps** on predictions made by GNNs to **identify relevant nodes** for a specific prediction; this provides valuable insights into the graph's topology and highlights the most important components
- Saliency Maps assign a score to each node in the network, which can be used to **rank paths** involving genes, pathways, diseases, and compounds

Loech J, Yang Y, Ekmei P, Dumontier M, and Celebi R, “Explaining Graph Neural Network Predictions for Drug Repurposing”, SWAT4HCLS 2024

Recal: Training Framework



Evaluation Metrics

Hits@K

- Counts the number of hits (correct predictions) within the top-K recommendations.
- A higher Hits@K indicates better accuracy in predicting relevant items

MR (Mean Rank)

$$MR = \frac{1}{N}\sum_{i=1}^{N} \text{Rank}_i$$

- Calculates the average position of correct predictions in the ranked order
- Lower MR values signify better model performance as correct predictions are closer to the top
- A float number greater than 0

MRR (Mean Reciprocal Rank)

$$MRR = \frac{1}{N}\sum_{i=1}^{N} \frac{1}{\text{Rank}_i}$$

- Evaluates the average reciprocal rank of correct predictions.
- Closer to 1 is better, between 0 and 1

Ground Truth: `[item3, item7, item1]`

Model Predictions (Ranked): `[ , , , , item3]`

Model Predictions (Ranked): `[ , , item7, , ]`

Model Predictions (Ranked): `[ , item1, , , ]`

N =3

$$MR = \frac{1}{3} \times (5 + 3 + 2) = \frac{10}{3}$$

$$MRR = \frac{1}{3} \times \left(\frac{1}{5} + \frac{1}{3} + \frac{1}{2}\right)$$

Results: Graph Neural Networks

- **GAT** achieves a **Hits@5 score** of **0.451** and a **Hits@10 score** of **0.672**
- **GraphSAGE** achieves a high **Recall Rate** of **0.992**

→ High emphasis on Recall in Drug Repurposing scenarios to minimize falsely predicted links between drugs and diseases:

Table 1
Results for predicting link (Compound, treats, Disease).

	GNN variant	Precision	Recall	Hits@5	Hits@10
DRKG	GraphSAGE	0.287	**0.992**	0.298	0.385
	GCN	0.361	0.871	0.280	0.409
	GAT	**0.834**	0.610	**0.451**	**0.672**

Explaining GNN Predictions for Drug Repurposing

- The proposed framework takes as input any prediction of a trained **GNN-based link prediction model** and returns an explanation in the form of a small subgraph of the input graph
- To return an explanation in the form of a small subgraph, we employed **Saliency Maps**, which is an attribution gradient-based method capable of attributing an importance score to the nodes and edges from the input graph via backpropagation

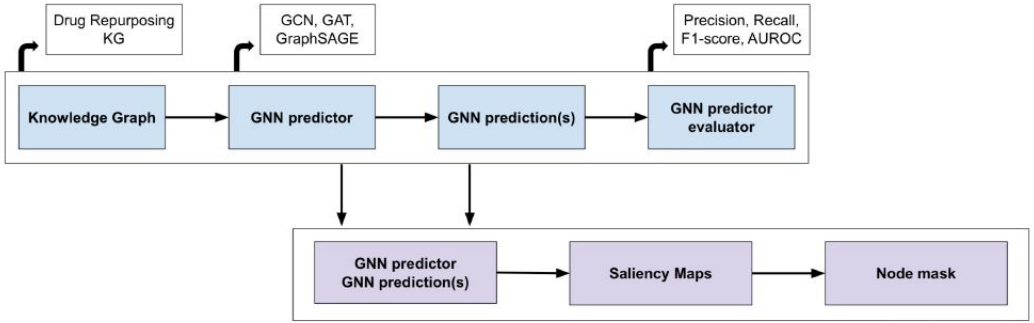


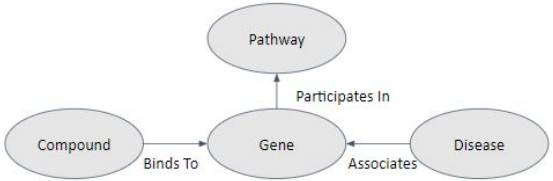
Figure 1: Pipeline of generating GNN prediction(s) and its node mask.

Loech J, Yang Y, Ekmei P, Dumontier M, and Celebi R, "Explaining Graph Neural Network Predictions for Drug Repurposing", SWAT4HCLS 2024

jloe2911 / ExplainableDrugRepurposing

Algorithm 1 Algorithm used to generate explanations.

function GENERATEEXPLANATORYSUBGRAPH(SM_scores, k) ▷ where *SM_scores* - scores derived from Saliency Maps, k - number of triples included in the explanation
    Let g_1, g_2, ..., g_n be ranked gene entities based on *SM_scores*
    RankedTriples = []
    for $g_i = 1$ to n **do**
        PathwayRel = ExtractRelations(g_i, "ParticipatesIn", "Pathway")
        DiseaseRel = ExtractRelations("Disease", "Associates", g_i)
        CompoundRel = ExtractRelations("Compound", "BindsTo", g_i)
        RankedPathwayRel = RankRelations(PathwayRel)
        RankedDiseaseRel = RankRelations(DiseaseRel)
        RankedCompoundRel = RankRelations(CompoundRel)
        RankedTriples.append(RankedPathwayRel[:k], RankedDiseaseRel[:k], RankedCompoundRel[:k])
    end for
    ExplanatorySubgraph = BuildExplanatorySubgraph(RankedTriples)
    return ExplanatorySubgraph
end function



Use Case: Donepezil treats Alzheimer

- The primary goal of Alzheimer's drugs, including Donepezil, is to **maintain elevated acetylcholine (ACh) levels**, thereby compensating for the loss of functioning cholinergic brain cells
- Figure 2 emphasizes the crucial role of **Donepezil binding to acetylcholinesterase (AChE)** and **butyrylcholinesterase (BChE)**

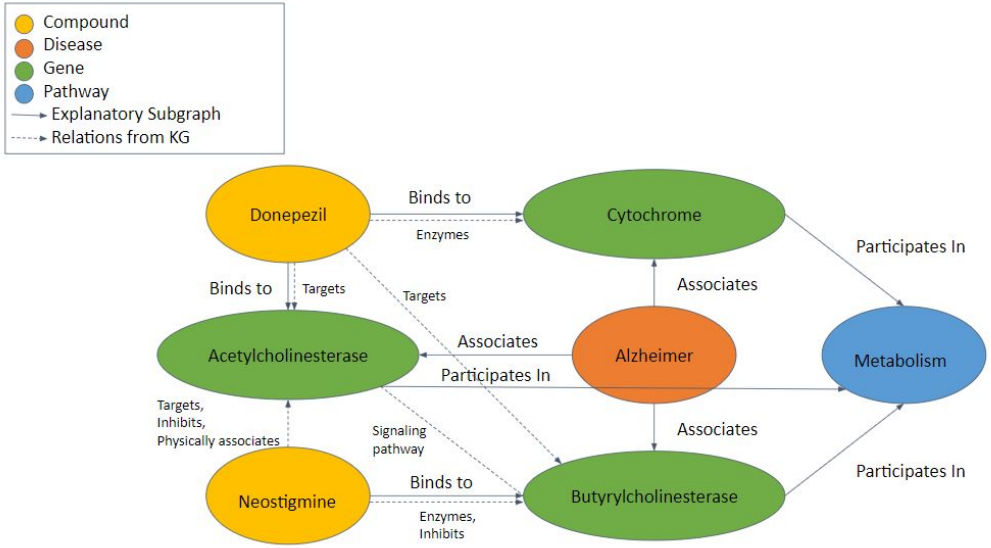


Figure 2: Explanatory subgraph for (Donepezil, treats, Alzheimer) including relationships from the original KG.

Use Case: Memantine treats Alzheimer

- Memantine, commonly prescribed for moderate to severe Alzheimer's disease, is believed to help prevent excess levels of the substance glutamate from damaging the brain
- Thus, according to our explainability framework and existing literature, **important genes associated with Memantine** include **glutamate receptor** and **acetylcholinesterase**

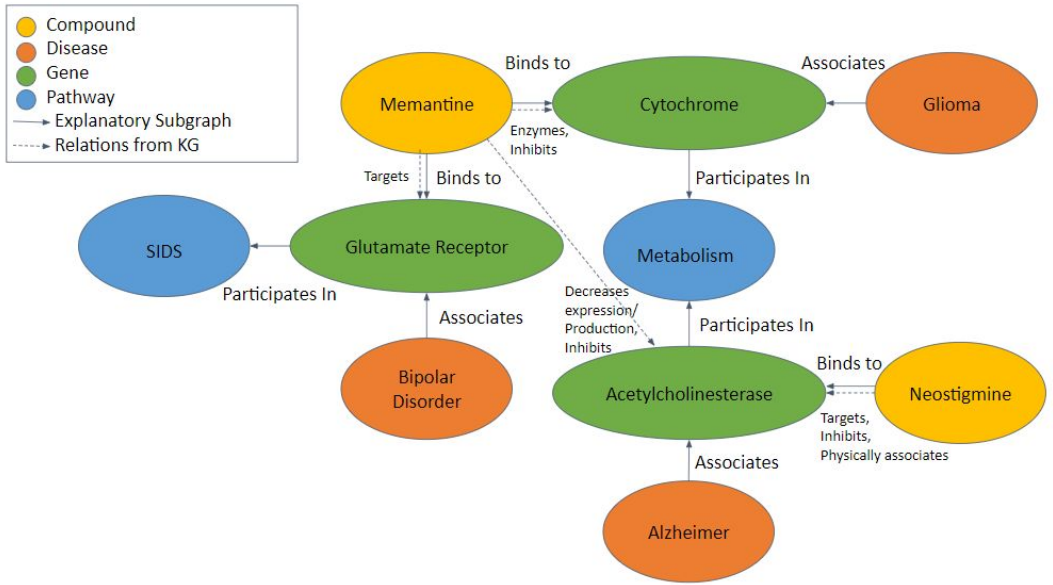


Figure 3: Explanatory subgraph for (Memantine, treats, Alzheimer) including relationships from the original KG.