Demystifying Graph Neural Networks

MANUEL DILEO

Connets LAB
Computer Science
UNIVERSITÀ DEGLI STUDI DI MILANO

Connets Lab

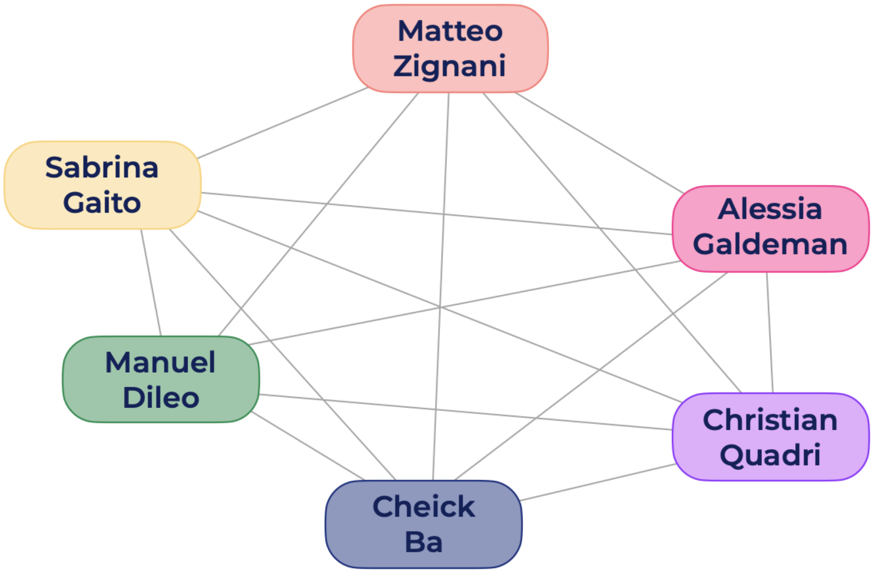
RESEARCH GROUP WORKS

Network evolution
Graph evolution rules
Change point/Anomaly detection

Graph Machine Learning
Social network analysis using GNN and LLM
Temporal Graph Learning for heterogeneous networks

User behaviour
Multilayer community detection
Influence of hubs in a user migration context
User strategies in a reward-based platform

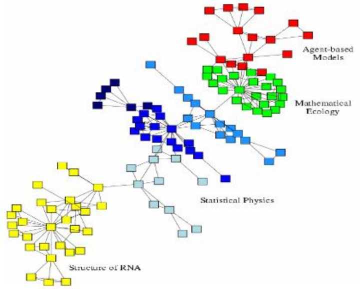
Web3 platform behavioral and network analysis
Blackchain-based online social network
NFT networks
Cryptocurrency networks (Luna, Steem, Ethereum, Sarafu)



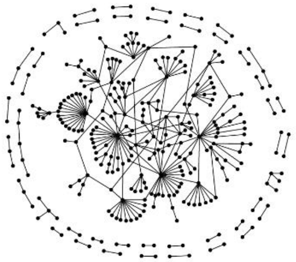
To see our works visit
https://connets.di.unimi.it/

Many Data are Networks

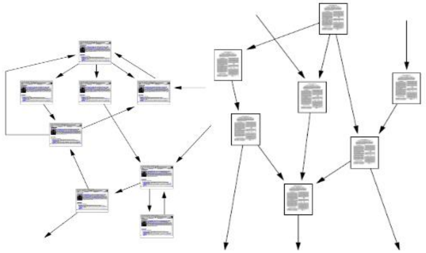
Social networks



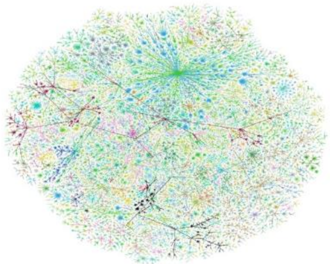
Economic networks



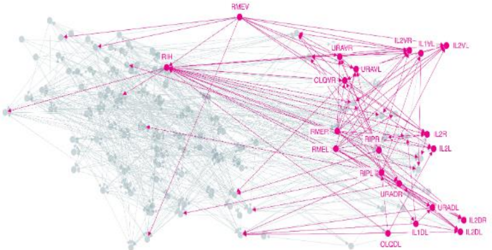
Biomedical networks



Information networks:
Web & citations



Internet



Networks of neurons

Graphs are an extremely powerful and general representation of data

ML task on networks

- Node classification
- Link prediction
- Community detection
- Graph classification

Classical ML task in networks

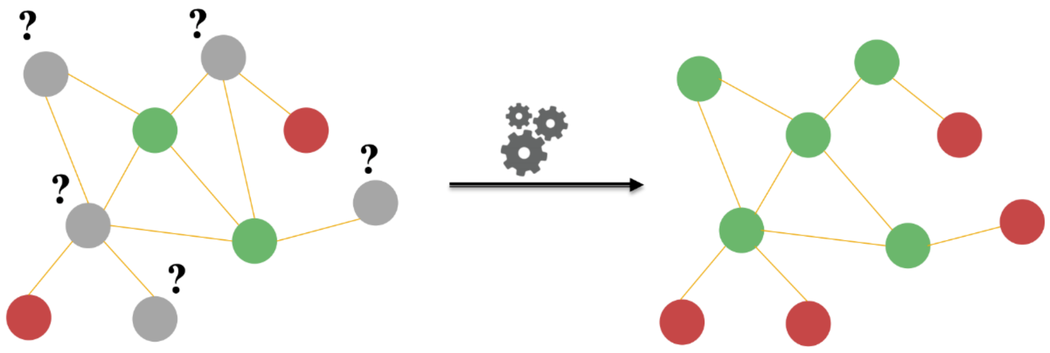
NODE CLASSIFICATION

Predict a label for a given node (e.g. the behaviour of an user in a social network)

Supervised or semi-supervised task

Network-approach as a way of improving prediction performance on classification tasks:

- Predict the topic of scientific paper
- Predict the genre of songs

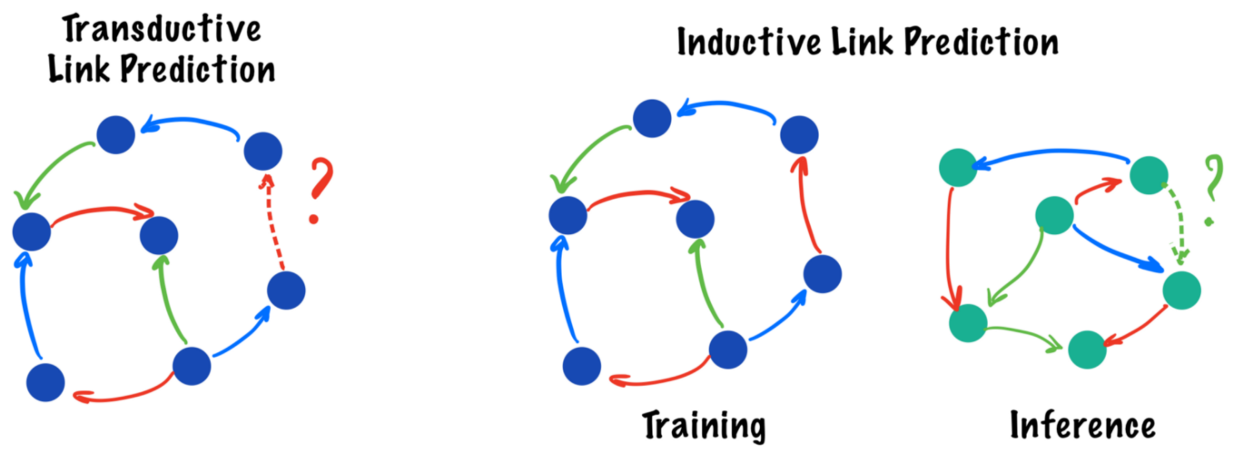


Classical ML task in networks

LINK PREDICTION

Predict wheter two nodes are linked (e.g. follow relations in a social network)

It can be treated as a binary classification task but it may lead to quite optimistic evaluation scenarios (Huang et al. , 2023)

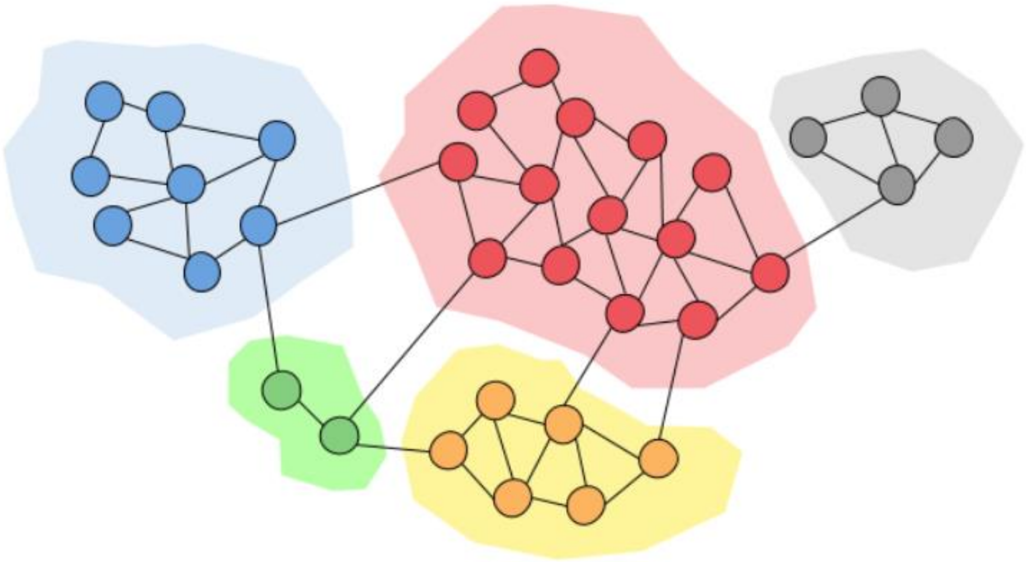


Classical ML task in networks

COMMUNITY DETECTION

Identify densely linked clusters of nodes

Unsupervised task

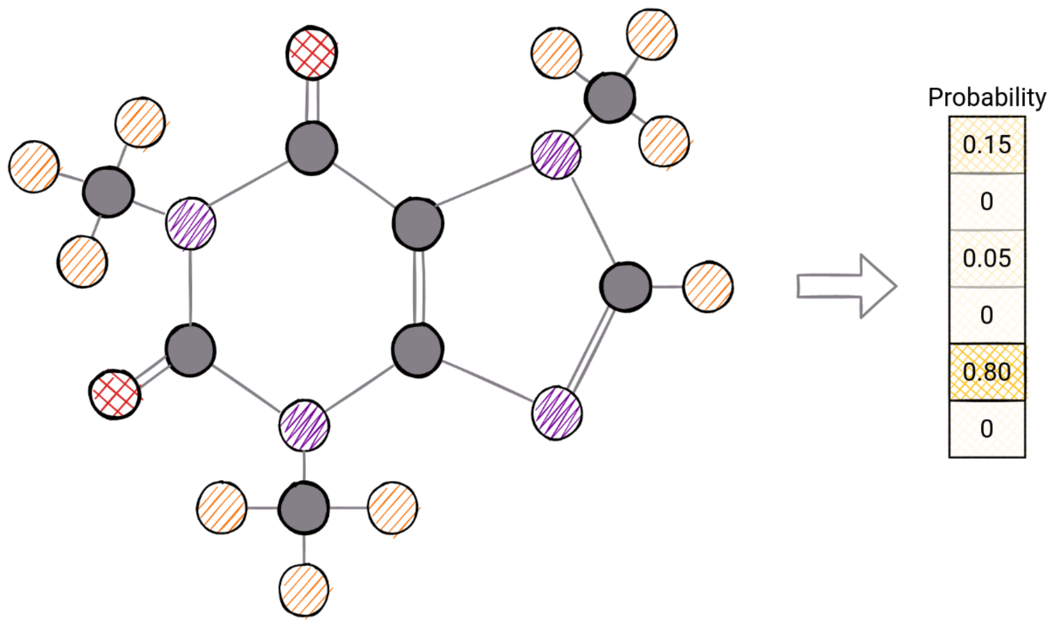


Classical ML task in networks

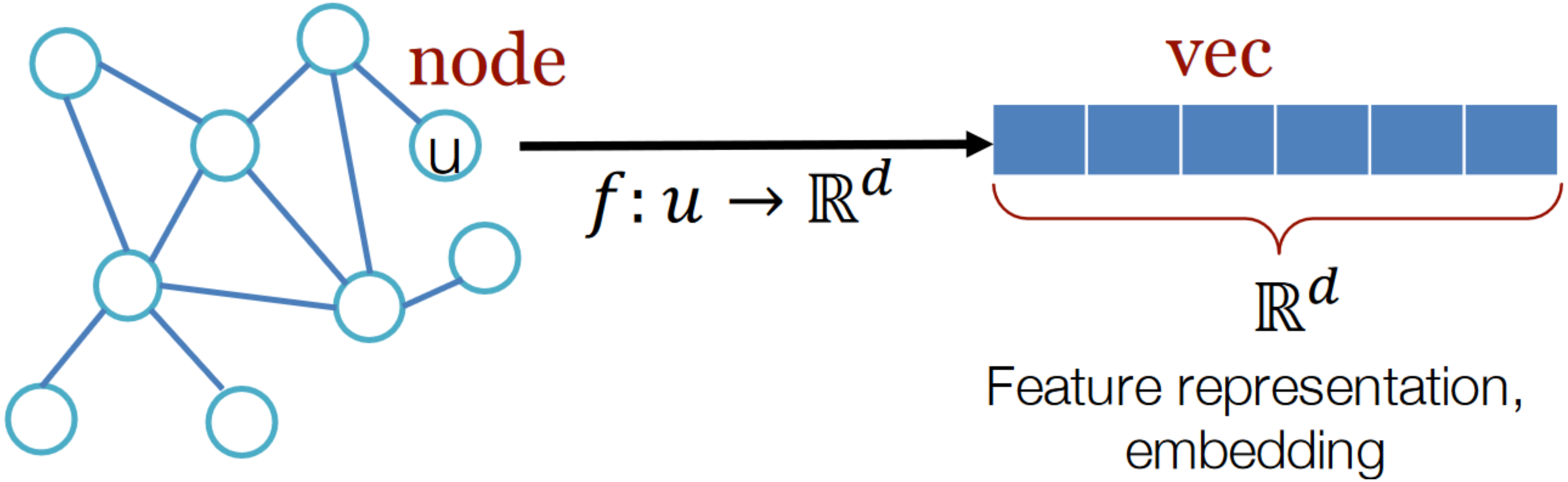
GRAPH CLASSIFICATION

Predict a label for a given graph in a dataset of graphs

- Disease associated to a certain brain network structure
- Role of a protein based on its molecular structure
- [...]

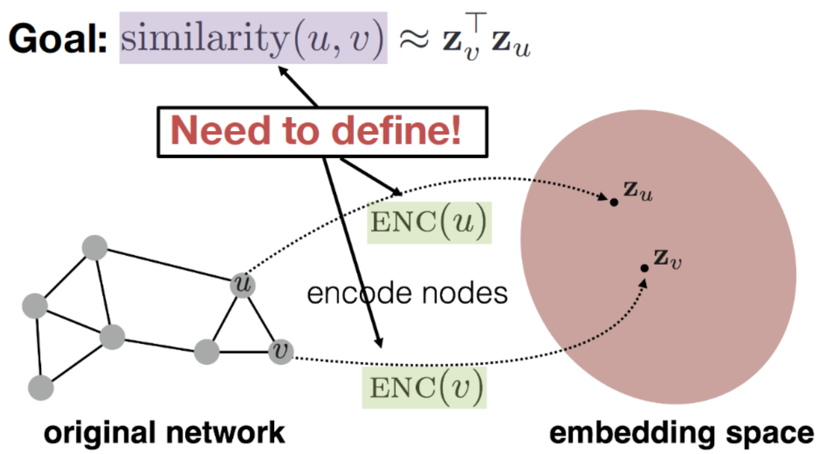


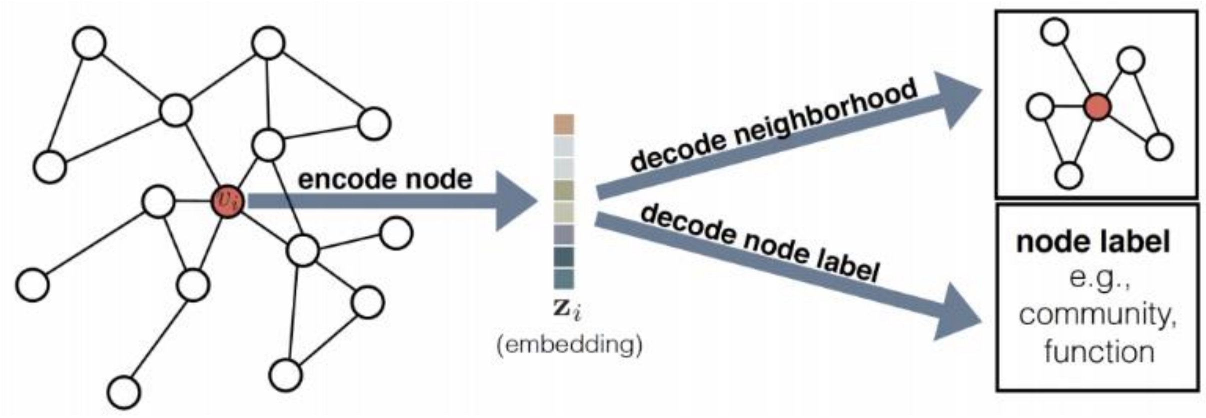
Graph embedding methods



The Encoder-Decoder Model (EDM)

- Similarity function
  - measures the similarity between nodes (can be omitted)
- Encoder function
  - generates the node embeddings
- Decoder function
  - Solve a downstream task using the node embeddings
- Loss function
  - checks the quality of the reconstruction



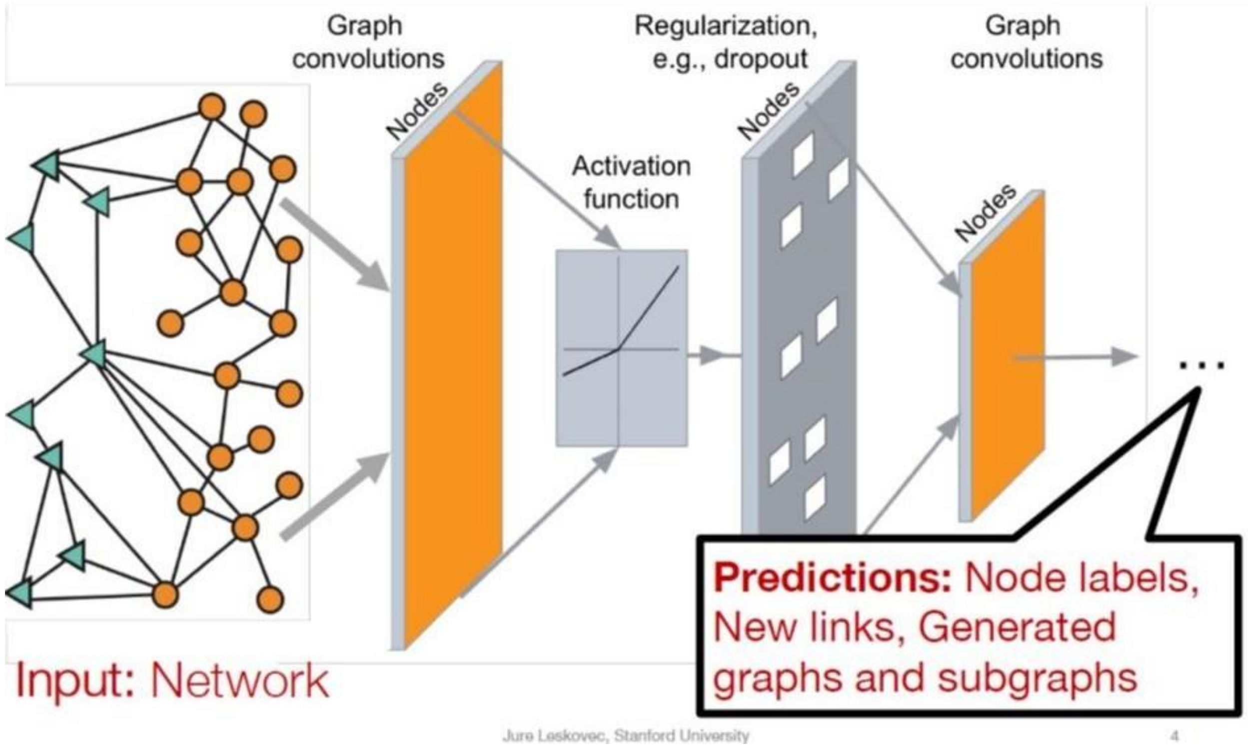


Challenges in graph computation

- Lack of consistent structure
- Node-order equivariance
  - Graphs often have no inherent ordering present amongst the nodes.
- Scalability
- Include node attributes
- Generalize on unseen nodes

Graph Neural Networks

- NNs that works naturally on graph-structured data.
- Automatic feature learning on graph with node attributes
- The encoder is a «complex» function that depends on the structure of the graph and the NN learnable parameters



Graph Neural Networks

FROM A PYTORCH POV

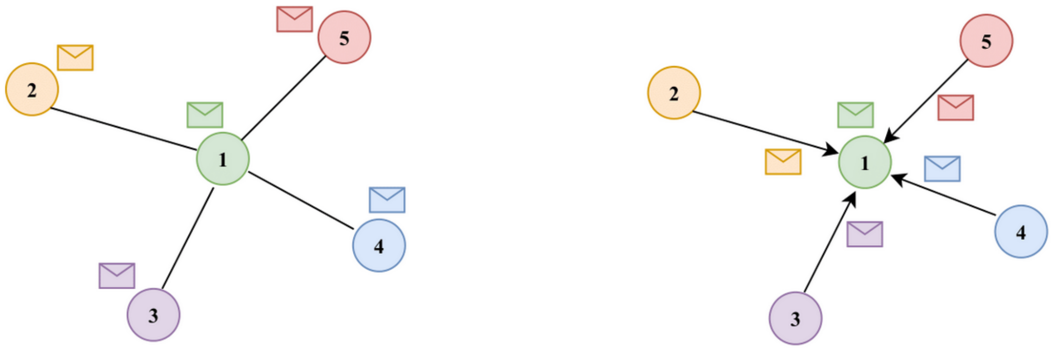
```python
from torch_geometric.nn import Linear
import torch.nn.functional as F

class MLP(torch.nn.Module):
    def __init__(self, input_dim, hidden_dim, output_dim):

        super(MLP, self).__init__()
        self.lin1 = Linear(input_dim, hidden_dim)
        self.lin2 = Linear(hidden_dim, output_dim)

    def forward(self, x, edge_index):
        h = self.lin1(x)
        h = F.relu(h)
        h = F.dropout(h, p=0.20)
        h = self.lin2(h)
        return h
```

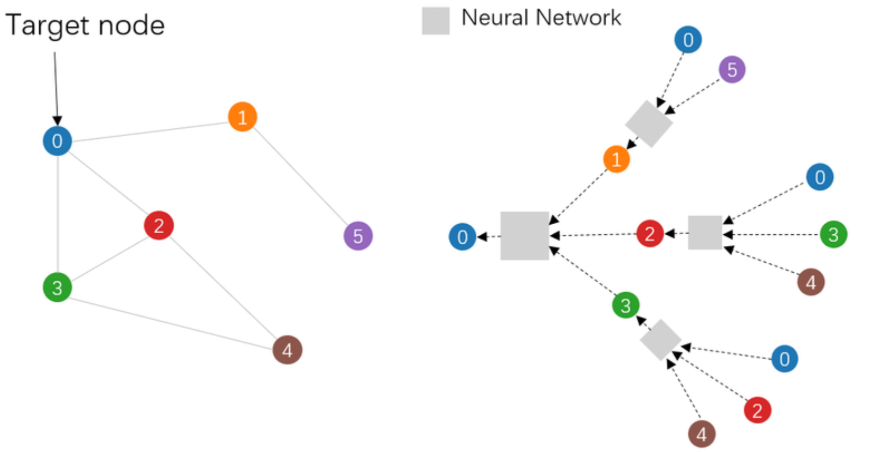
```python
from torch_geometric.nn import GCNConv
import torch.nn.functional as F

class GNN(torch.nn.Module):
    def __init__(self, input_dim, hidden_dim, output_dim):

        super(GNN, self).__init__()
        self.conv1 = GCNConv(input_dim, hidden_dim)
        self.conv2 = GCNConv(hidden_dim, output_dim)

    def forward(self, x, edge_index):
        h = self.conv1(x, edge_index)
        h = F.relu(h)
        h = F.dropout(h, p=0.20)
        h = self.conv2(h, edge_index)
        return h
```

Graph Neural Networks

THE MESSAGE PASSING FRAMEWORK

- 1-hop message passing: each node sends its features to its neighbors
- New node features as a combination of original features and an aggregation of the features of the neighborhood





(a) Input graph

(b) Neighborhood aggregation

Graph Neural Networks

POPULAR GNN LAYERS

We obtain different GNN layers considering different combination and aggregation functions.

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

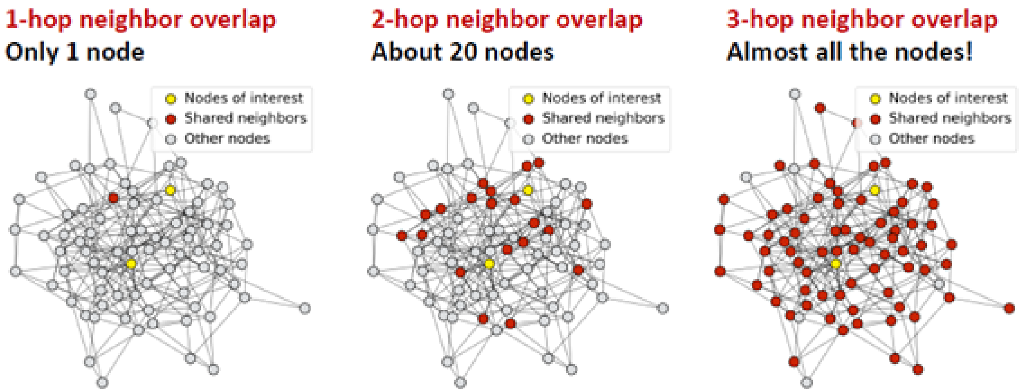
Challenges in graph computation

- Lack of consistent structure
- Node-order equivariance
- Scalability
  - Parameter-sharing: re-use the same weights for all the nodes
  - GraphSAGE: Sample neighborhood
- Include node attributes
- Generalize on unseen nodes

Stacking GNN layers

- The depth influences the «receptive field»
- Oversmoothing problem: all the node embeddings converge to the same value
  - Especially true for small-world networks

- Receptive field overlap for two nodes
  - The shared neighbors quickly grows when we increase the number of hops (num of GNN layers)

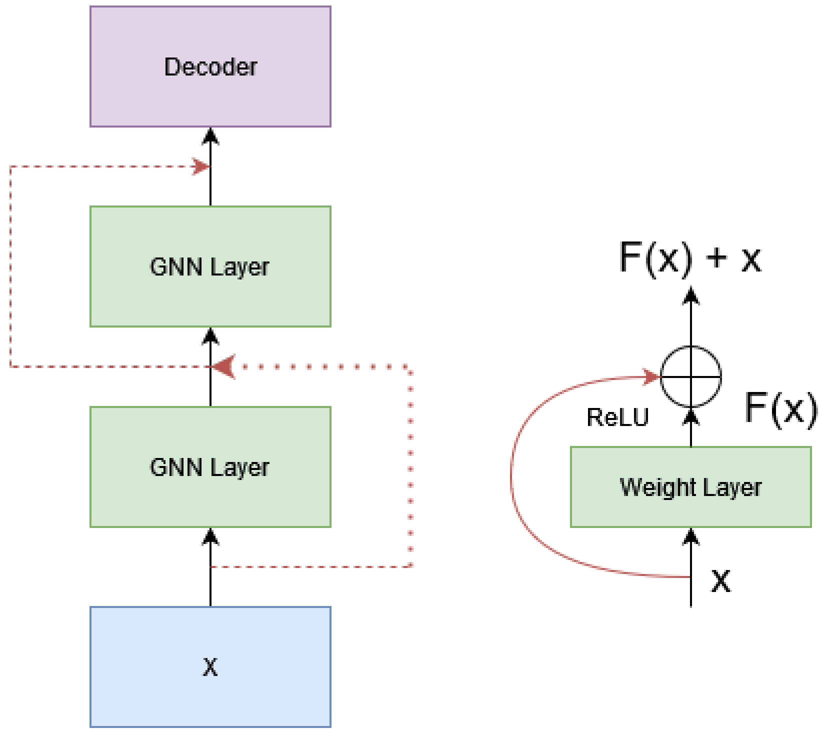


High depth ≠ high expressiveness. The latter relies on the design of single layers and computational graphs

Mitigate oversmoothing

A survey on oversmoothing in GNNs (Rusch et al., 2023)

- Avoid using more than 3 GNN layers
- Penalize solutions that lead to oversmoothing with explicit regularization terms
- DropEdge: implicit regularization by adding noise to the opt process
- Skip Connections: do not forget the initial node features

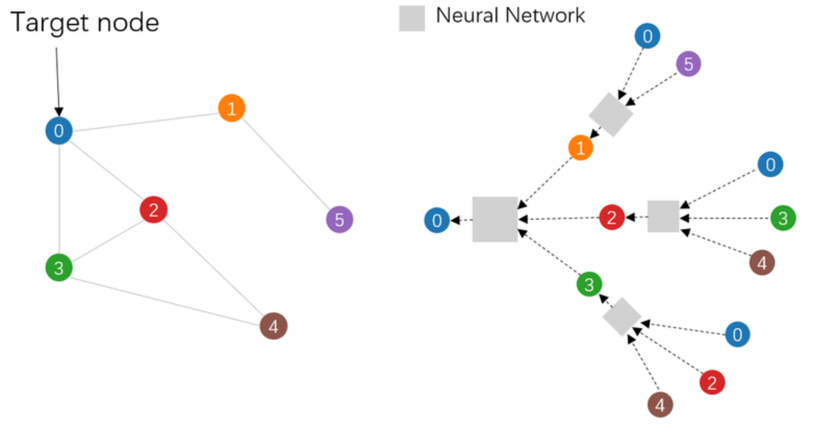


On the other side… over-squashing

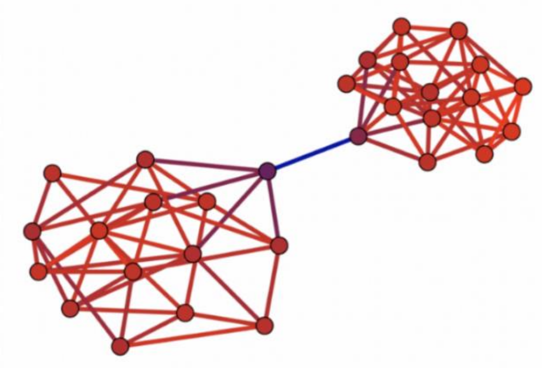
One-hop message passing implies that node features are insensitive to information contained at distant nodes.

GNNs may be not the best solution for long-range tasks (consider Matrix Factorization or Random-Walk based methods)

Oversquashing can arise from specific network patterns like bottlenecks



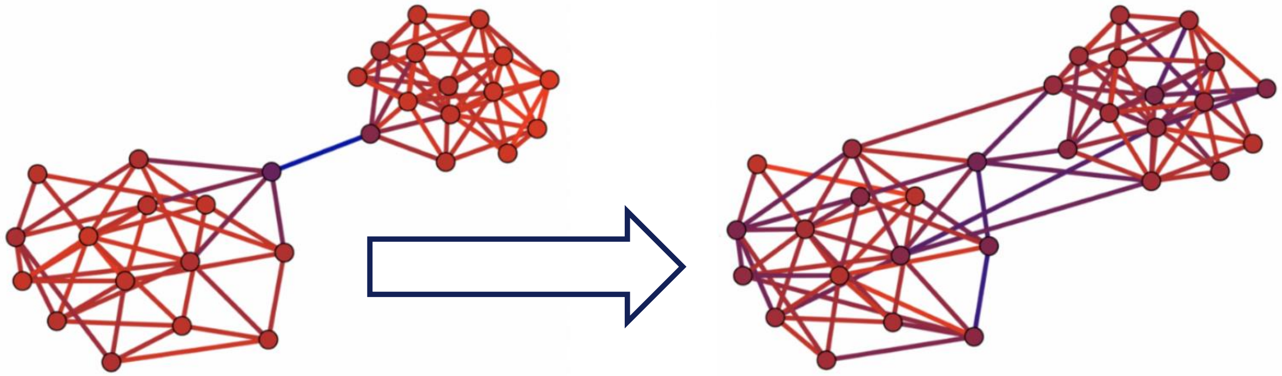
(a) Input graph (b) Neighborhood aggregation



Mitigate over-squashing

On Over-Squashing in Message Passing Neural Networks: The Impact of Width, Depth, and Topology ([Di Giovanni et al., 2023](#))

- Increasing NN width or depth does not really mitigate oversquashing
- Consider X-hop message passing
- Graph topology plays the greatest role, consider Graph Rewiring techniques



Are GNNs always a good solution?

DISCLAIMER: WE REFER TO «STANDARD» GNNS
WITH 1-HOP MESSAGE PASSING

Node Classification

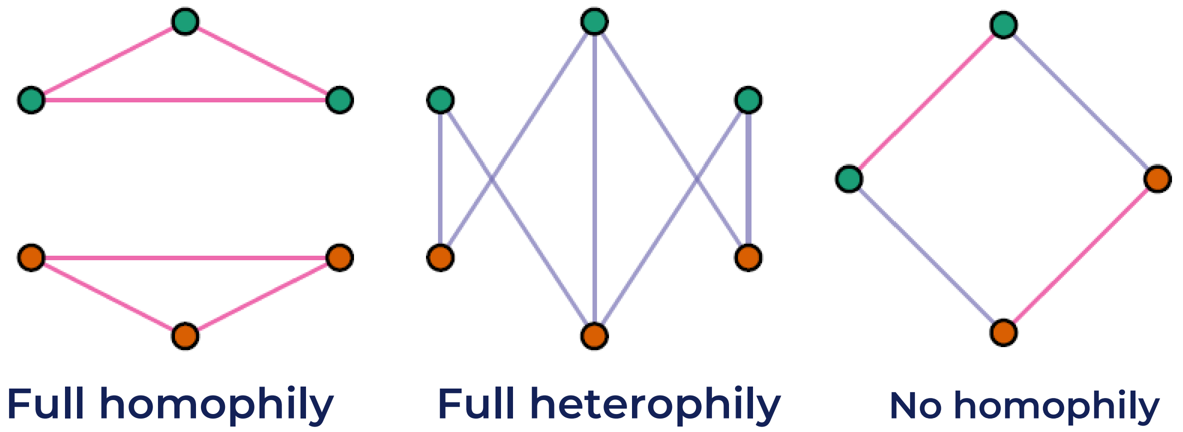
HOMOPHILY VS HETEROPHILY

Task: Predict node labels. **Solution**: softmax on the output layer

Homophily: a link between nodes with the same label occurs at a higher rate than among nodes with different labels.

GNNs struggle with heterophilic networks: they are mixing node attributes of dissimilar nodes!

Homophilic network without node features?
Use Label Propagation



Use GNNs for homophilic networks with node features

Node classification in heterophilic networks

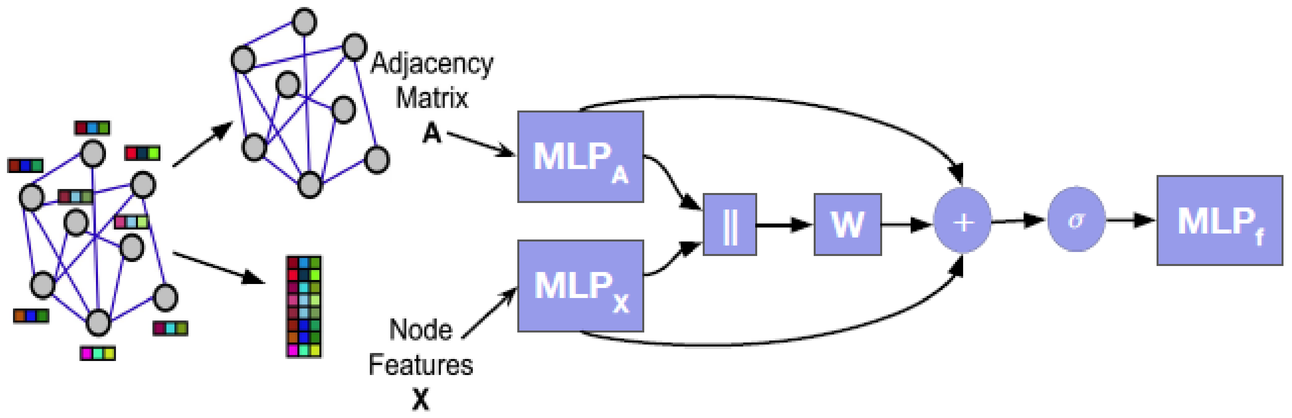
LINKX (NEURIPS 2021)

Examples of homophilic networks:

- Social Networks on user interests
- Citation Networks on topics

Examples of heterophilic networks:

- Social Networks on gender
- Citation Networks on year of publication
- Subgraph of Wikipedia (same topic) on page views (more in general, degree).



Key idea: give strong importance to node features!

Link Prediction

LINK PREDICTION IS A 2-ORDER TASK

Task: Predict missing/future links.

Solution: Scoring function on «candidate pairs embedding»
E.g. (a,b) in E? Sigmoid(gnn(a) * gnn(b))

Factorization-based models often outperform GNNs on **transductive** tasks ([Chen et al.,2022](#))

Two links that involves **symmetric** but different nodes cannot be distinguished by «vanilla» GNNs

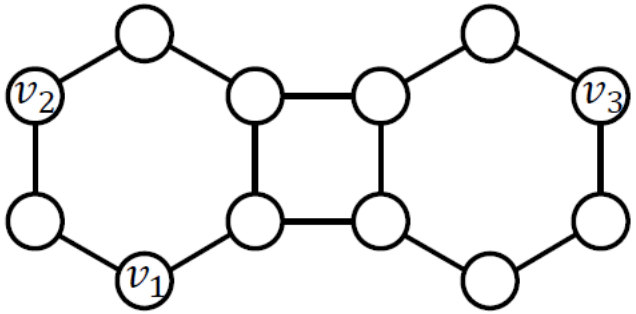


Figure 1: The structural roles of link (v_1, v_2) and link (v_1, v_3) are different, but GAE will assign equal probabilities to them.

GNNs without node features cannot distinguish link structural roles

Revisiting GNNs for Link Prediction

[LABELING TRICK (NEURIPS 2022)]

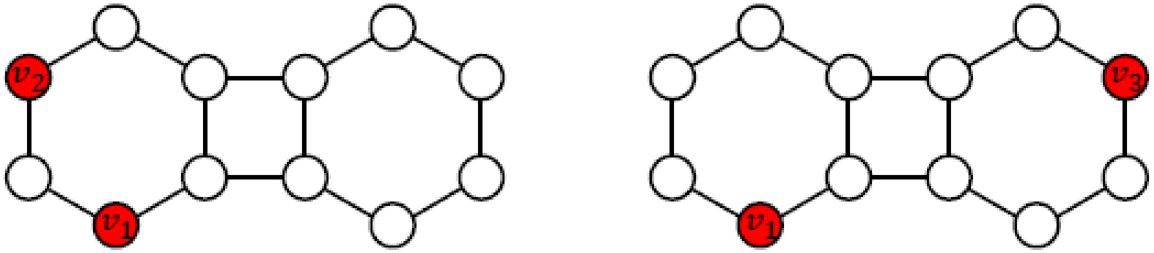


Figure 2: When we predict (v_1, v_2), we will label these two nodes differently from the rest, so that a GNN is aware of the target link when computing v_1 and v_2's embeddings. Similarly, when predicting (v_1, v_3), nodes v_1, v_3 will be labeled differently. The aggregated embedding of v_1, v_2 in the left graph will be different from the aggregated embedding of v_1, v_3 in the right graph, enabling GNNs to predict (v_1, v_2) and (v_1, v_3) differently.

Key idea: create fake features to distinguish edges!

In practical scenarios, nodes have features -> v2-v3 are distinguishable

No features? Random Features Strengthen Graph Neural Networks (Sato et al., 2021)

Graph Classification

1-WL IS ALMOST ALL YOU NEED

Task: Predict graph labels. Solution: graph pooling on node embeddings. Context: no features

Graph isomorphism problem: distinguish if two graphs are «structurally the same» (exist a mapping of the nodes that preserve all the edges). 1-WL test necessary but insufficient condition for graph isomorphism.

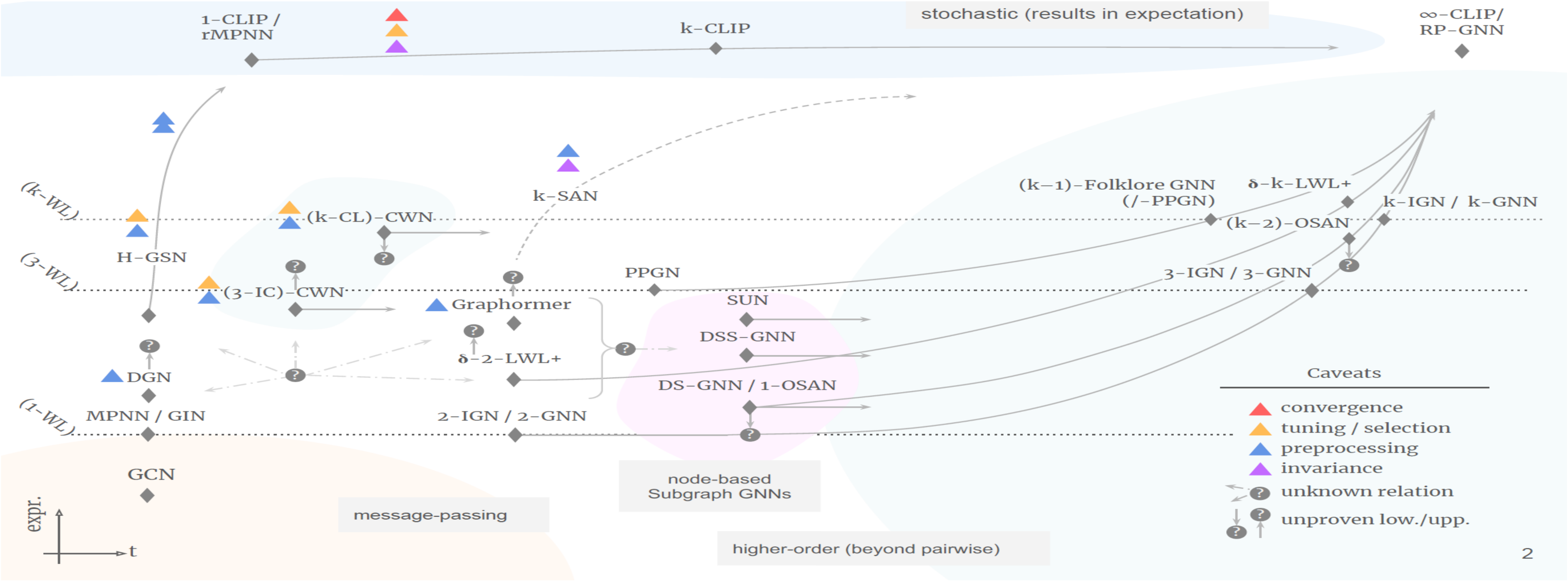
Theorem (Morris et al.): any GNN's expressive power is upper bounded by the 1-WL in terms of distinguishing non-isomorphic graphs.

!! If two non-isomorphic graphs detected as isomorphic by 1-WL have different labels, they cannot be distinguished !!

Luckily, GNNs are not limited by their expressiveness in practice (Zopf et al., 2022)

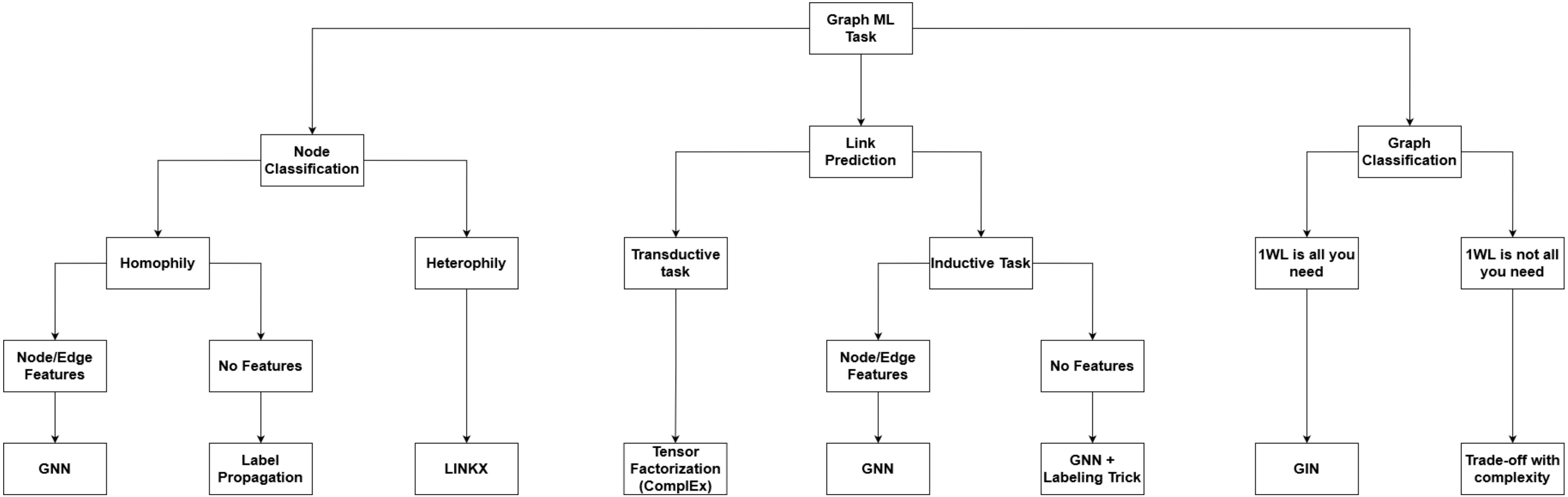
Vanilla GNNs are almost expressive as the 1-WL test for graph isomorphism, but in most of the cases is all you need

TUTORIAL: EXPLORING THE PRACTICAL AND THEORETICAL LANDSCAPE OF EXPRESSIVE GRAPH NEURAL NETWORKS



Find a good compromise between expressiveness and computational complexity on your dataset

My GraphML map



SPAM

- Paper accepted at the Temporal Graph Learning workshop at NeurIPS 2023! Preprint available [“DURENDAL: graph deep learning framework for temporal heterogeneous networks”](#).
- Paper accepted at Machine Learning Journal! The paper “Temporal Graph Learning for Dynamic Link Prediction with Text in Online Social Networks” will be available online soon.

Suggested material on GraphML

- http://web.stanford.edu/class/cs224w/
- https://www.cs.mcgill.ca/~wlh/grl_book/
- https://openreview.net/forum?id=BkxSmlBFvr
- https://distill.pub/2021/understanding-gnns/
- https://www.youtube.com/watch?v=ASQYjbUBYzs
- t.me/graphML

Thanks to Jure Leskovec and Michael Bronstein for some schemas taken from their slides

References

- S. Huang et al., “Temporal graph benchmark for machine learning on temporal graphs”, in Advances in Neural Information Processing Systems, 2023.
- T. K. Rusch et al., “A Survey on Oversmoothing in Graph Neural Networks”, 2023.
- F. Di Giovanni et al., “On over-squashing in message passing neural networks: The impact of width, depth, and topology”, in International Conference on Machine Learning, 2023.
- D. Lim et al., “Large Scale Learning on Non-Homophilous Graphs: New Benchmarks and Strong Simple Methods,” in Advances in Neural Information Processing Systems, 2021.
- Y. Chen et al., “ReFactor GNNs: Revisiting Factorisation-based Models from a Message-Passing Perspective,” 2022.
- Zhang et al., “Labeling Trick: A Theory of Using Graph Neural Networks for Multi-Node Representation Learning,” in Advances in Neural Information Processing Systems, 2021.
- R. Sato et al., “Random Features Strengthen Graph Neural Networks,” in SDM, 2021.
- C. Morris et al., “Weisfeiler and Leman go Machine Learning: The Story so far,” 2022.
- M. Zopf, “1-WL Expressiveness Is (Almost) All You Need”, 2022.

Thanks for your attention