Better Explainability with Graphs

What drives Graph Neural Network predictions

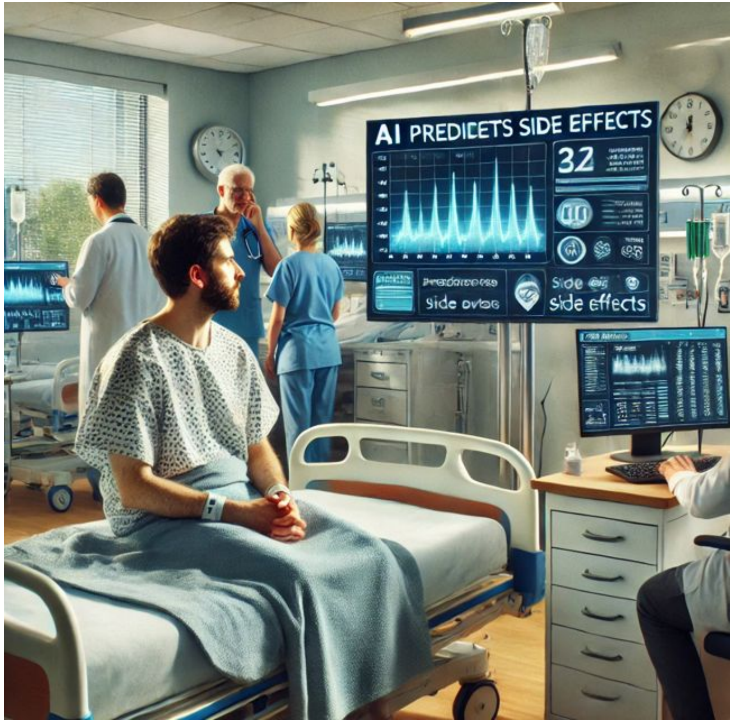
by Raphael von Lottner, Shan Jiang and Johannes Jones

This is Tom.

Tom is sick and needs medication.

AI does not like Toms medication.

What would you do?



Explainability matters!

WHY did the AI predict side effects?

What is Explainable AI (xAI)?

Making **machine learning models** and their **predictions** understandable for humans.

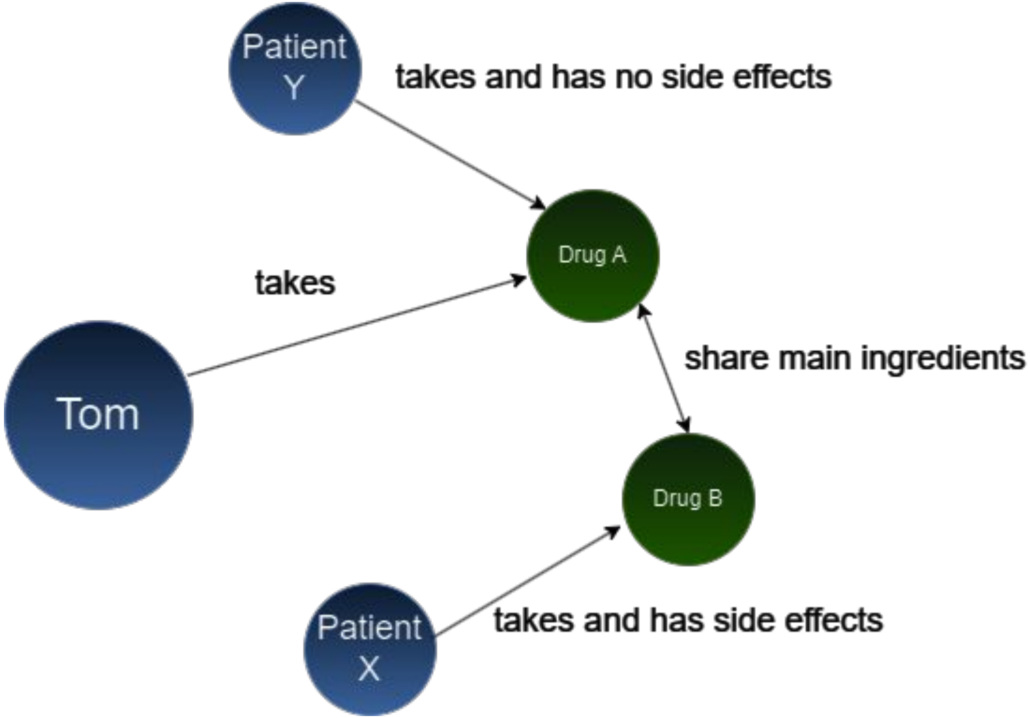
- Trust
- Compliance (GDPR)
- Ethics

We use graphs because they perform well and can be nicely visualized for xAI.

What are Graphs anyway?

Why Graphs are important?

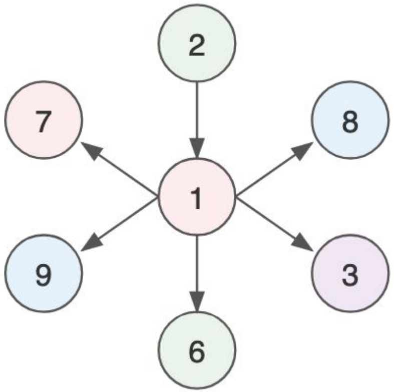
- Natural representation of relational data
- Preserve structural information
- Enable complex pattern recognition



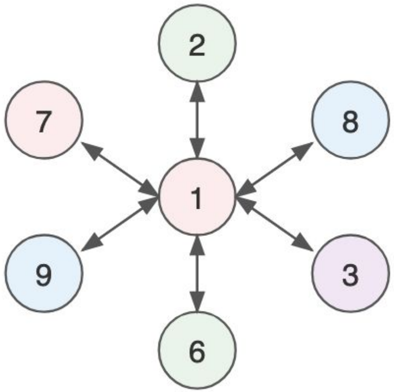
What are Graphs anyway?

- Node classification
- Link prediction

Directed



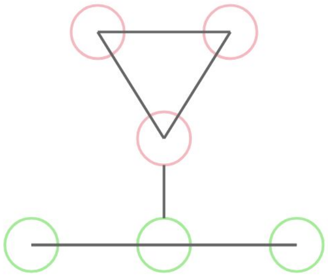
Undirected



Node classification task: Homophilic vs Heterophilic

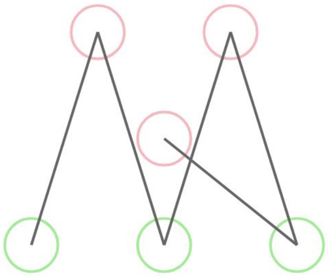
- Nodes connect to similar nodes.
- Nodes connect to dissimilar nodes.

Homophilic



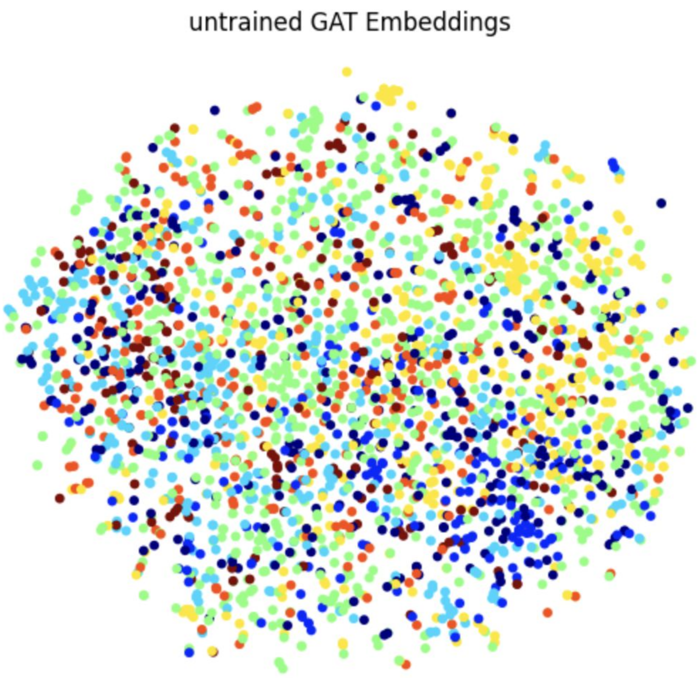
Similar nodes connect to each other
(e.g., friends with similar interests)

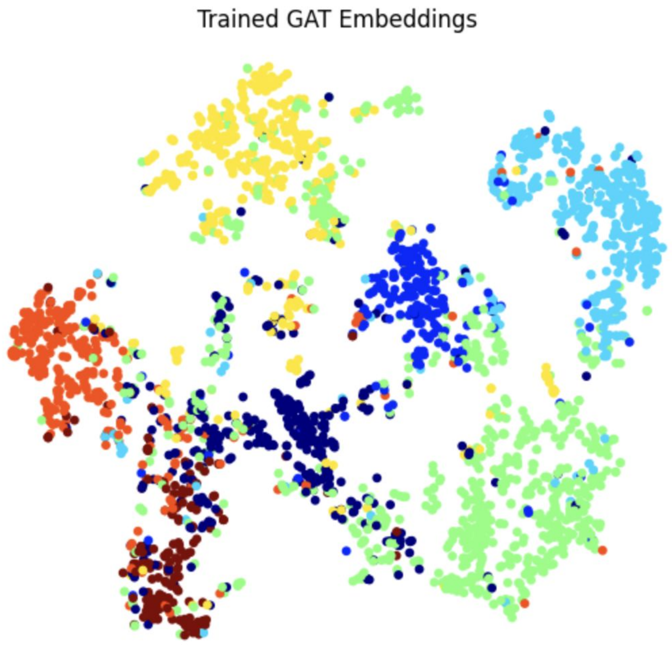
Heterophilic



Different nodes tend to connect
(e.g., buyer-seller relationships)

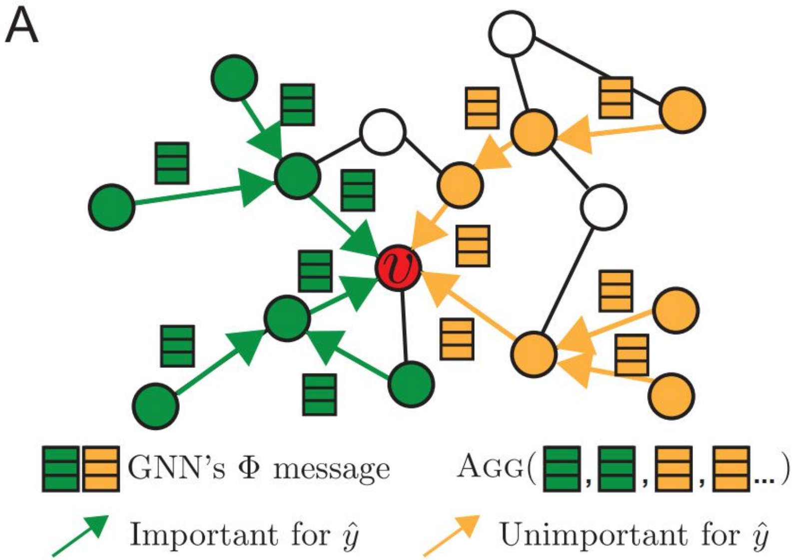
Node feature embeddings in 2D before and after training where different colors belong to different classes





GNNExplainer

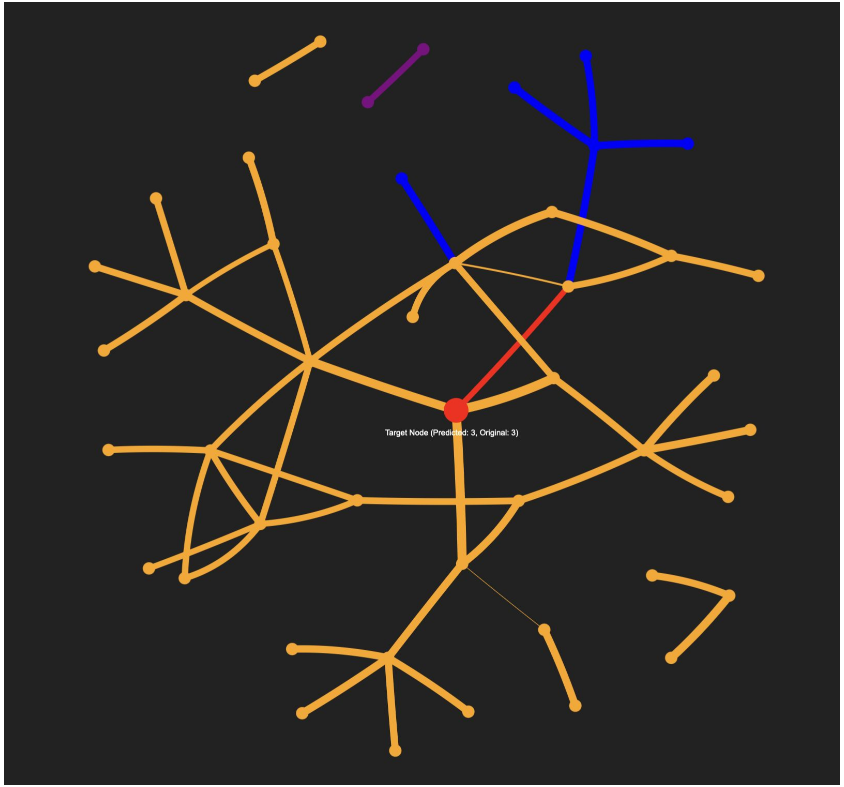
Explanation: optimization on masks (node, feature, edges) in order to extract a subgraph that contributes to a graph neural network's prediction.



Ying, Zhitao, et al. (2019)

GNNExplainer

Subgraph visualization:
edge_mask > threshold 0.5



Limitations:

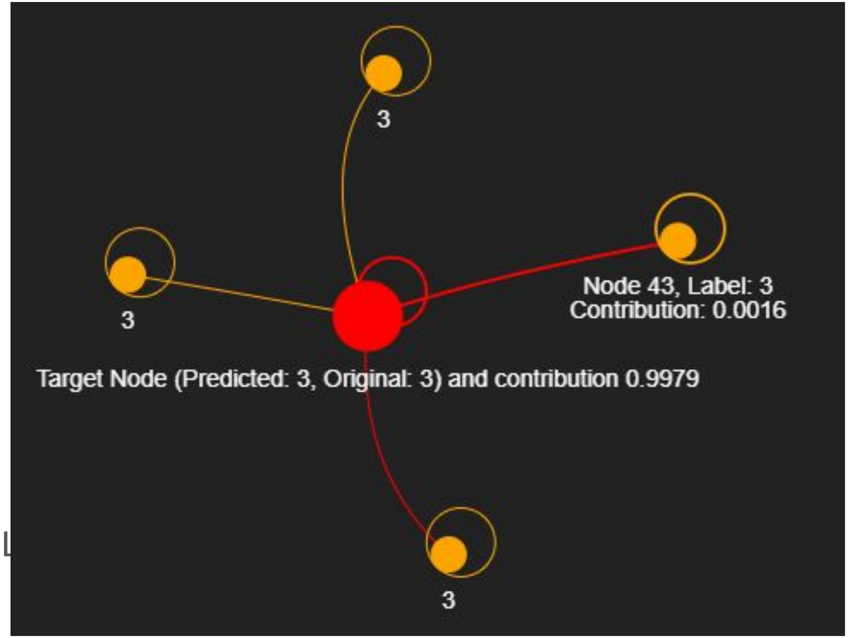
- better suitable for homophilic graphs
- sensitivity to hyperparameters
- lack of explainability metrics
- black-box nature of optimization

ATTENTION!!

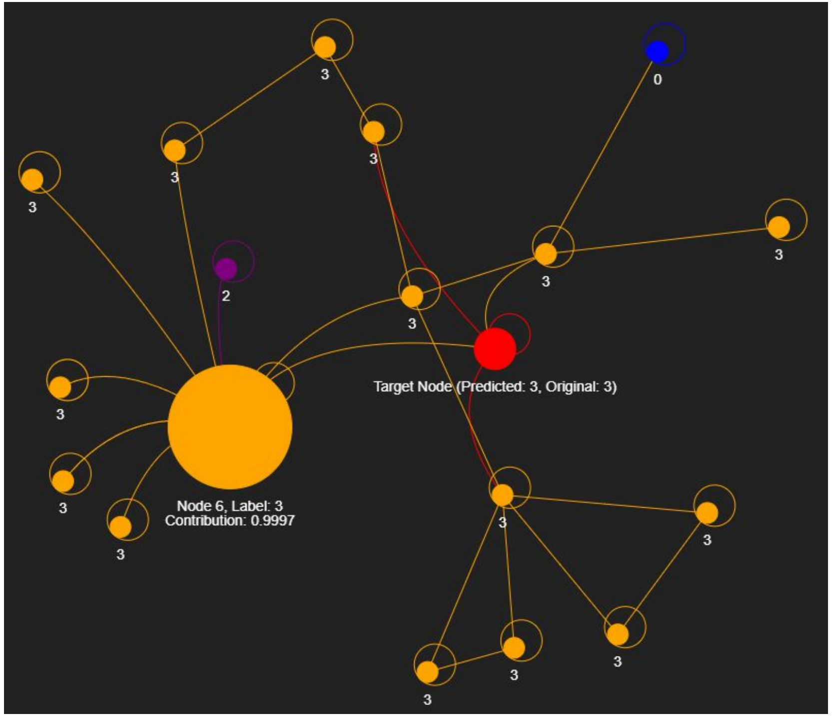
Like in Transformers…

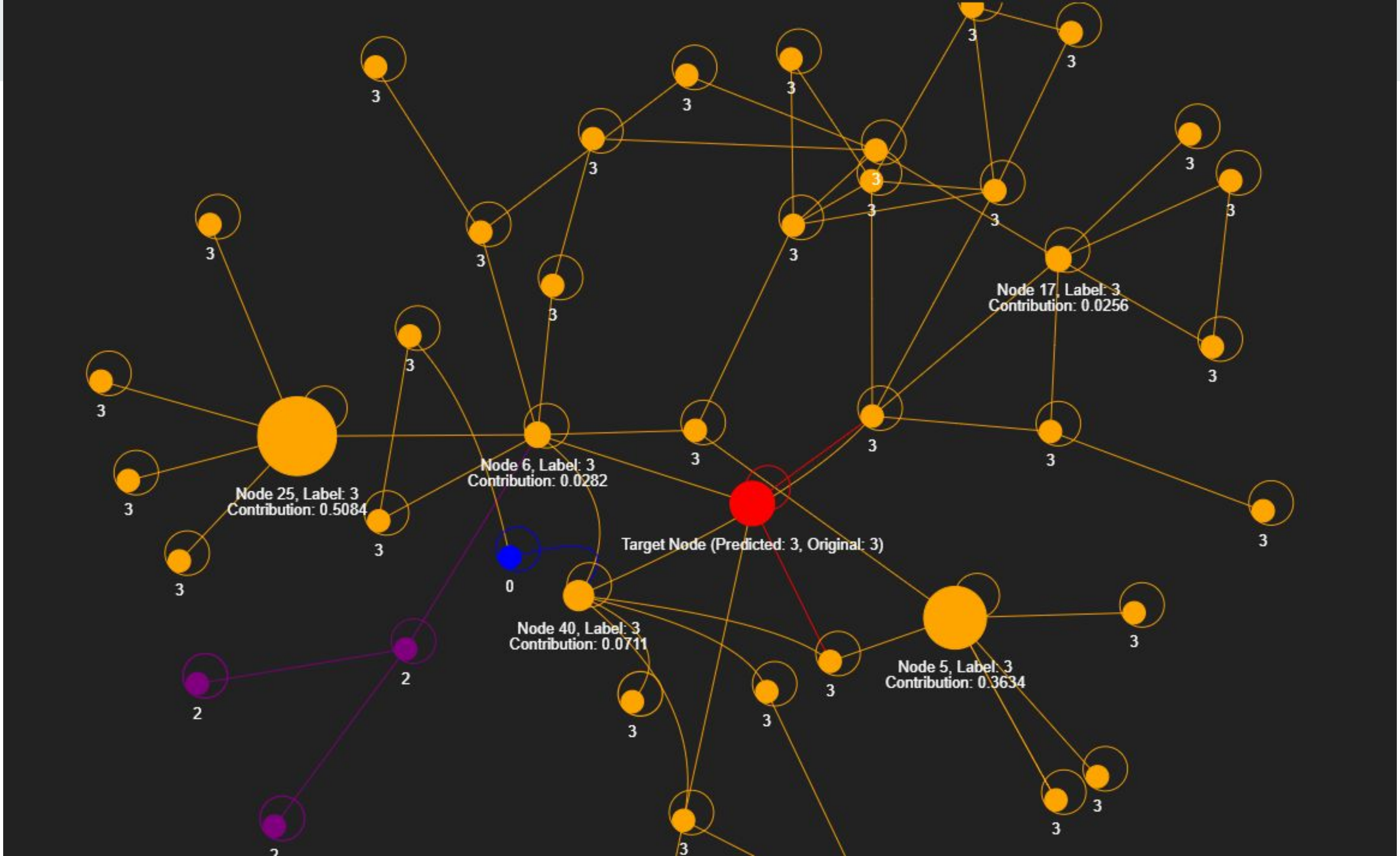
Let's make it more technical!

Layer 1:

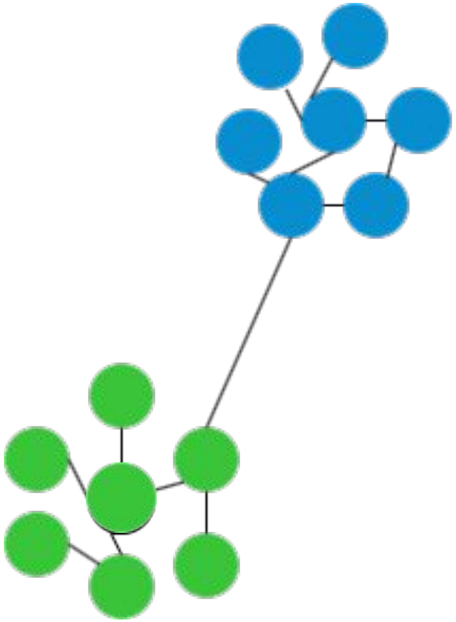


Layer 2:

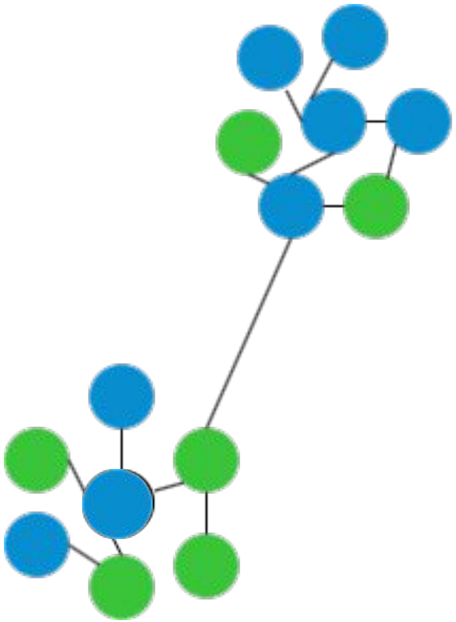



Node 17, Label: 3
Contribution: 0.0256
Node 25, Label: 3
Contribution: 0.5084
Node 6, Label: 3
Contribution: 0.0282
Target Node (Predicted: 3, Original: 3)
Node 40, Label: 3
Contribution: 0.0711
Node 5, Label: 3
Contribution: 0.3634

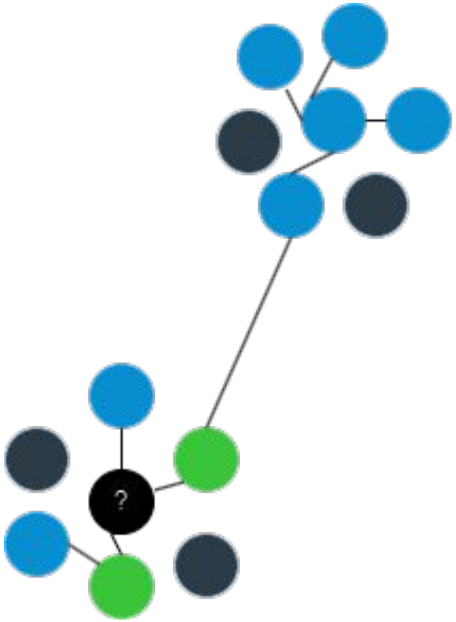
2 kinds of Graphs



Homophilic



Heterophilic



New Topology (Graph structure)

CoGNN, a brand new gnn... published in 2024...

arXiv:2310.01267v2 [cs.LG] 09 Jun 2024

Cooperative Graph Neural Networks

Ben Finkelshtein Xingyue Huang Michael Bronstein

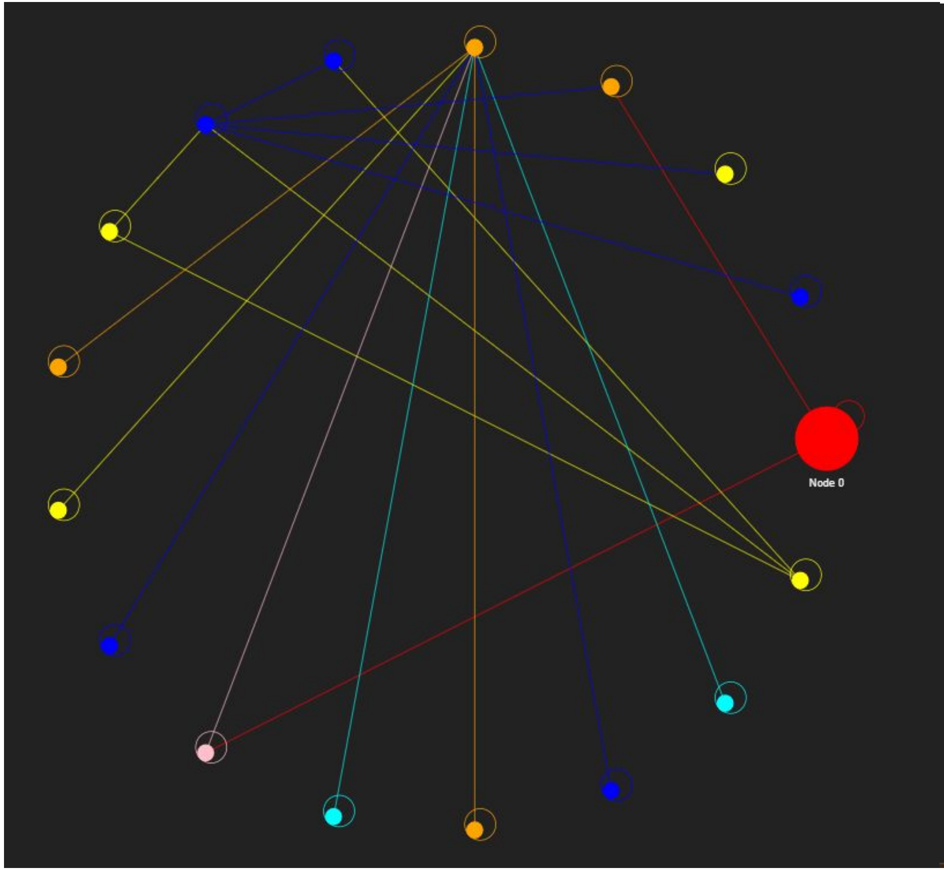
İsmail İlkan Ceylan

Abstract

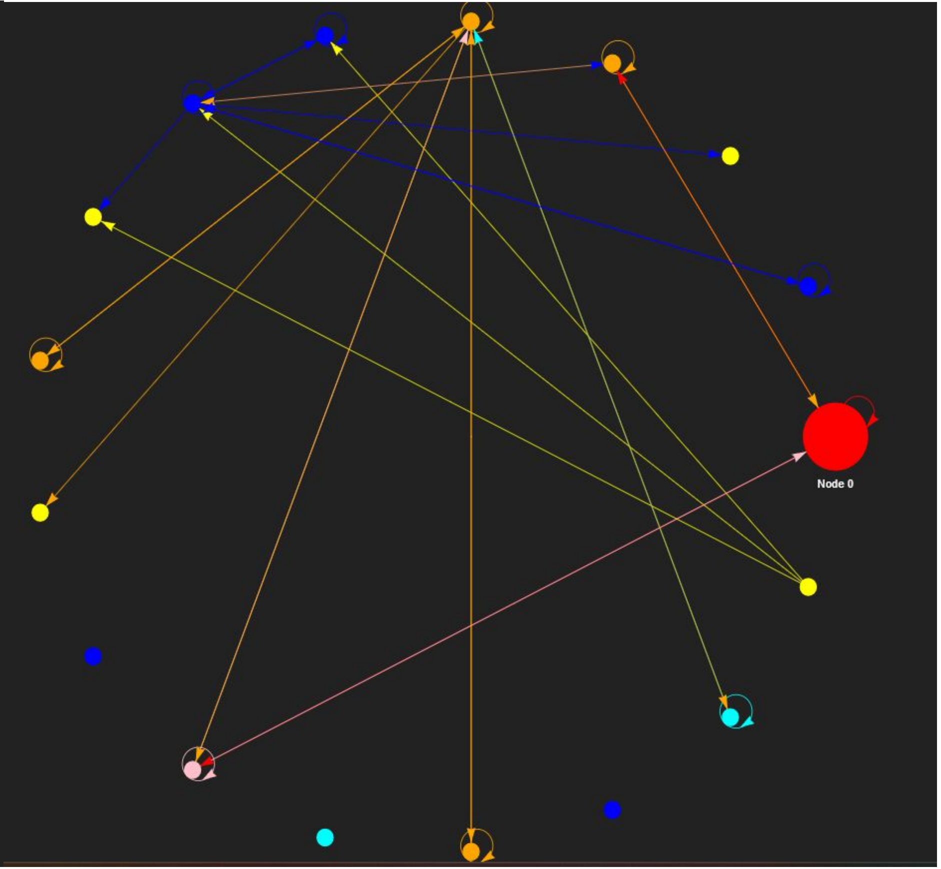
Graph neural networks are popular architectures for graph machine learning, based on iterative computation of node representations of an input graph through a series of invariant transformations. A large class of graph neural networks follow a standard message-passing paradigm: at every layer, each node state is updated based on an aggregate of messages from its neighborhood. In this work, we propose a novel framework for training graph neural networks, where every node is viewed as a *player* that can choose to either 'listen', 'broadcast', 'listen and broadcast', or to 'isolate'. The standard message propagation scheme can then be viewed as a special case of this framework where every node 'listens and broadcasts' to all neighbors. Our approach offers a more flexible and dynamic message-passing paradigm, where each node can determine its own strategy based on their state, effectively exploring the graph topology while learning. We provide a theoretical analysis of the new message-passing scheme which is further supported by an extensive empirical analysis on synthetic and real-world data.

graph neural networks, dynamic message passing, information flow

Original Graph on a homophilic dataset

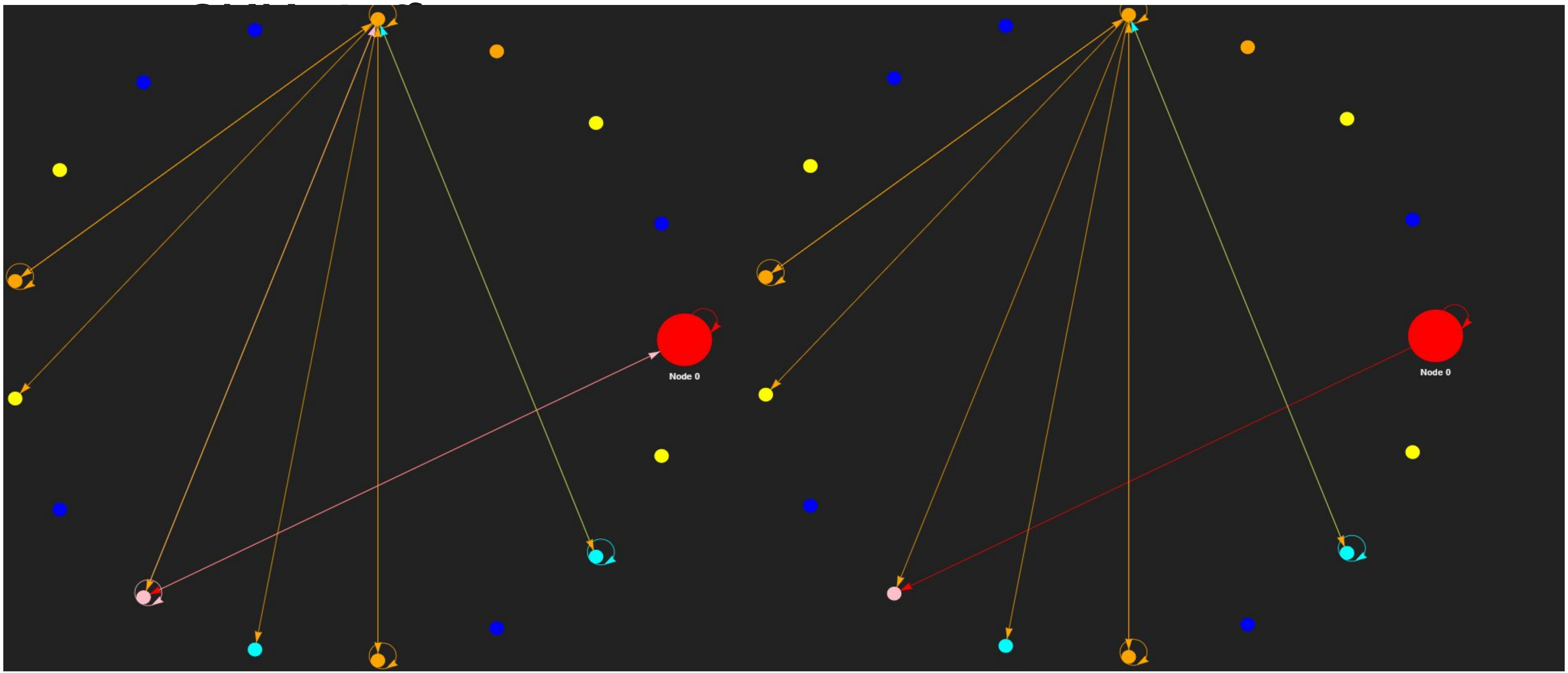


Layer 1



CoGNN

Layer 2 | Layer 3



Summary

Key Takeaways

- Graphs are powerful data structures with powerful models
- Explainability is different for different stakeholders
- Explainability is difficult but necessary

Thank you very much for your attention!

Model Architecture Limitations of GNNs and GAT

- Over-squashing: Information loss in long-range dependencies

    Cause: Trying to put too much information into a single feature vector

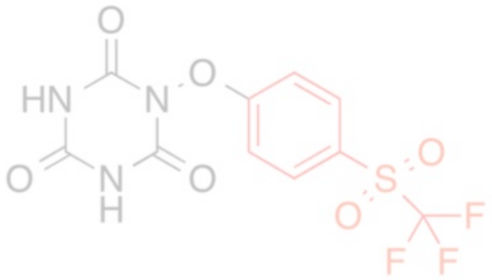
- Over-smoothing: Node features become too similar

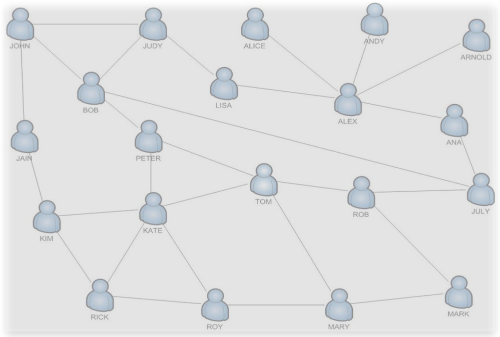
    Cause: Updating a vector too often

- Fixed topology during message passing
- Difficulty in explainability

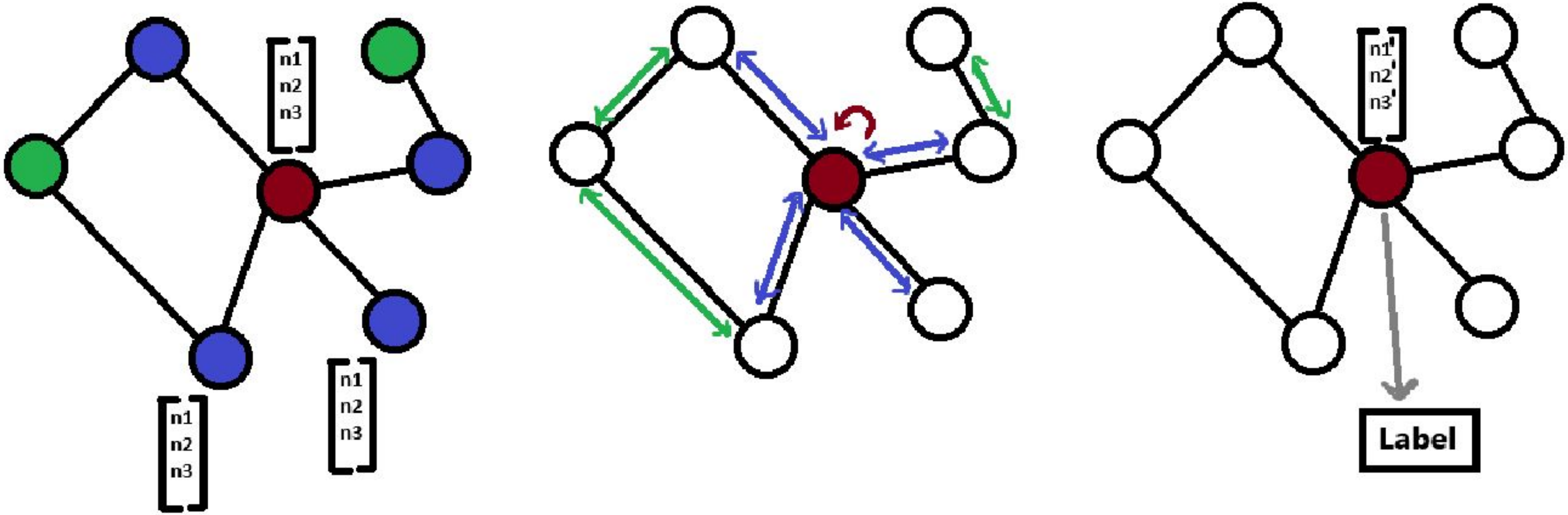
Where Applications of Graphs in industry

- Fraud detection
- Customer segmentation
- Biomedical applications
- Recommendation systems





GNN Architecture for node predictions



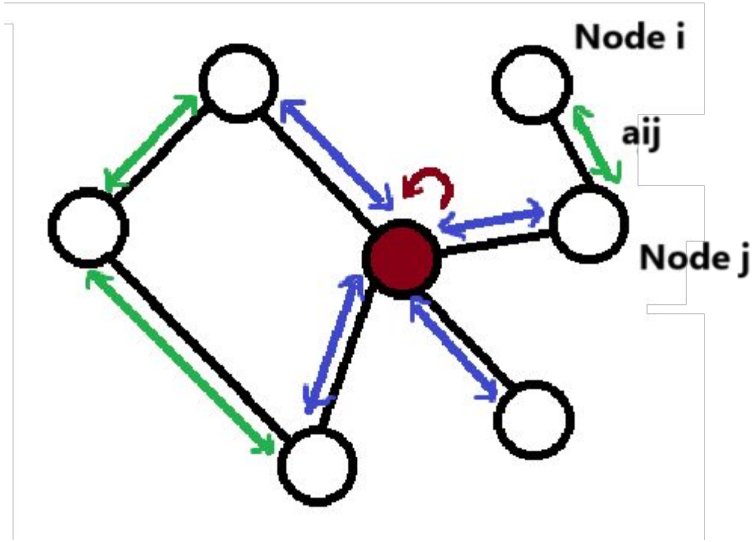
Graph with node features

Feature Aggregation

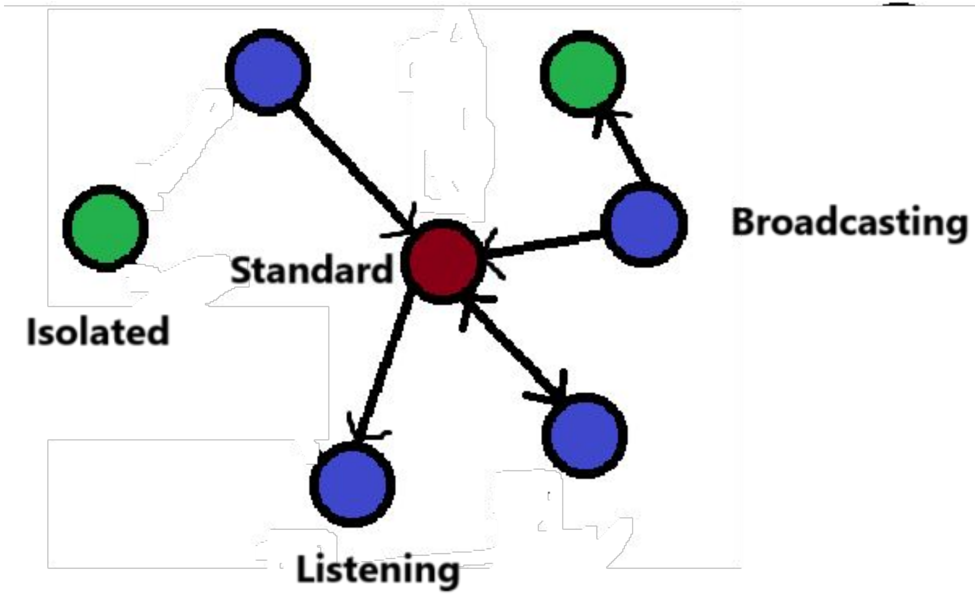
Final Feature with prediction

GAT and CoGNN Model architecture

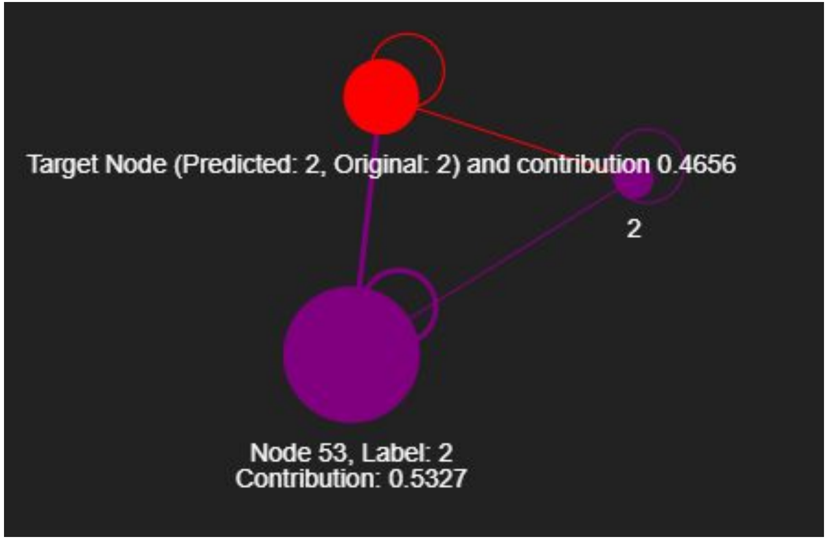
GAT



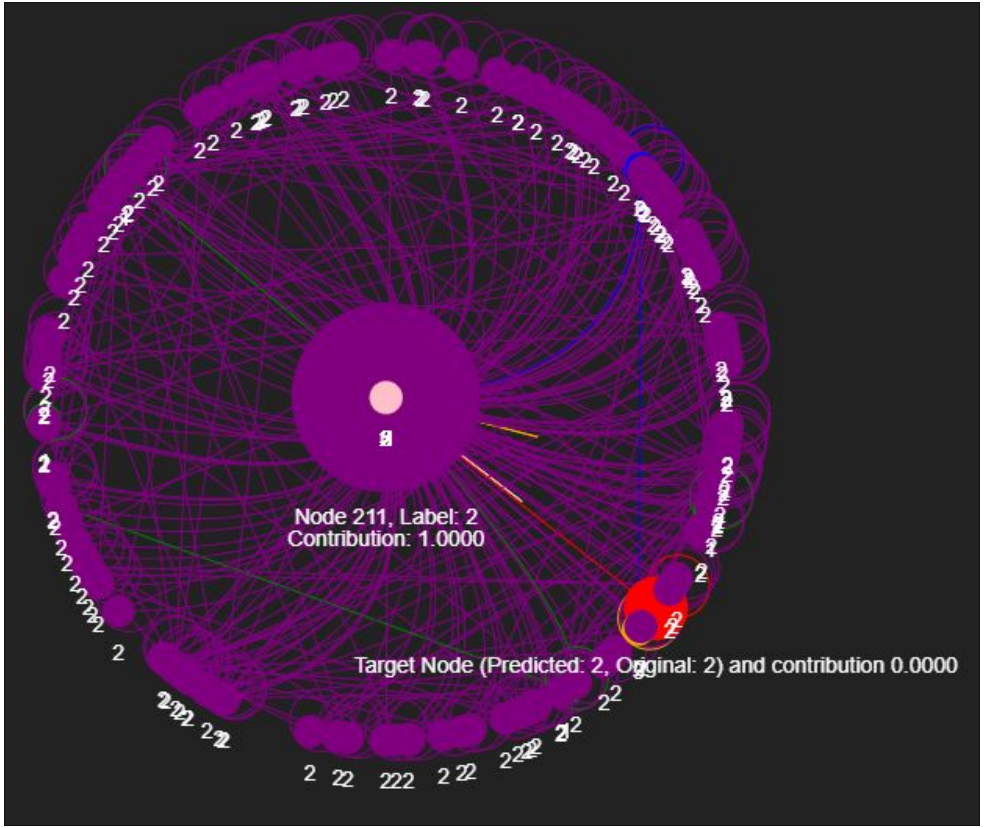
CoGNN



Layer 1:

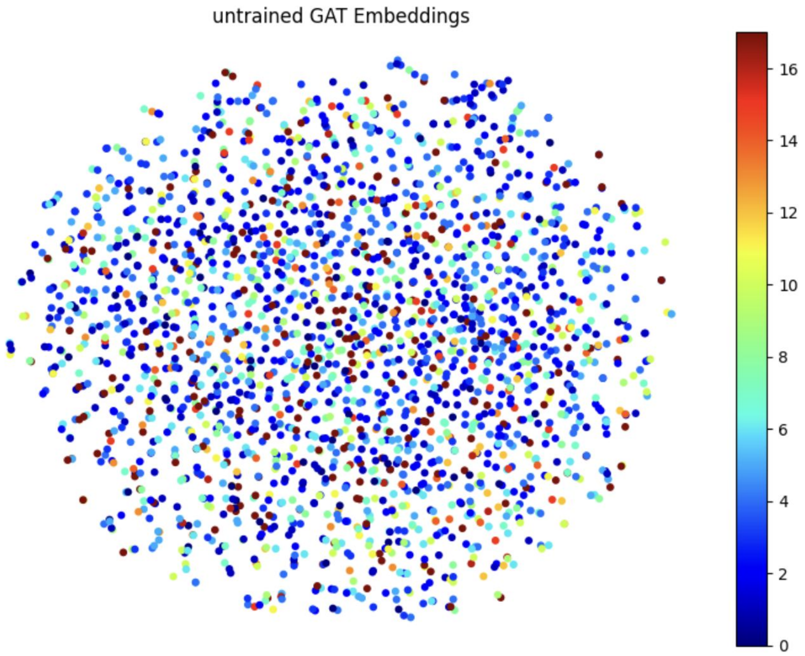


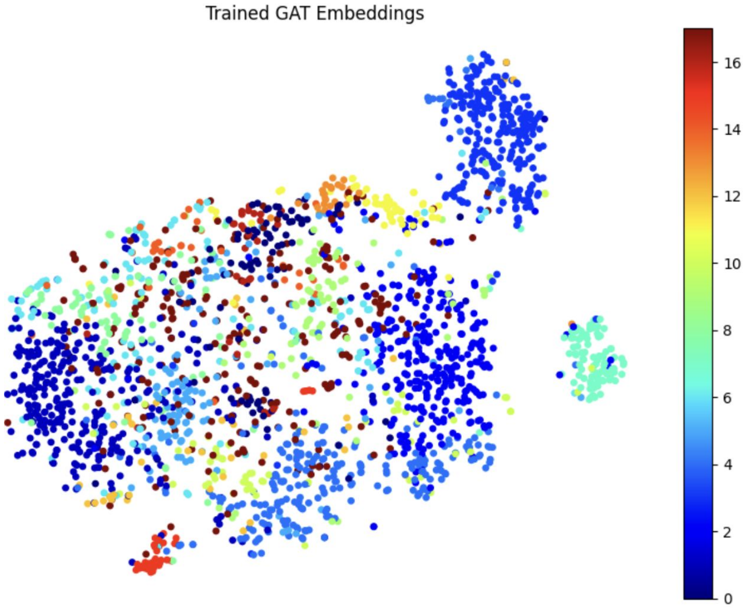
Layer 2:

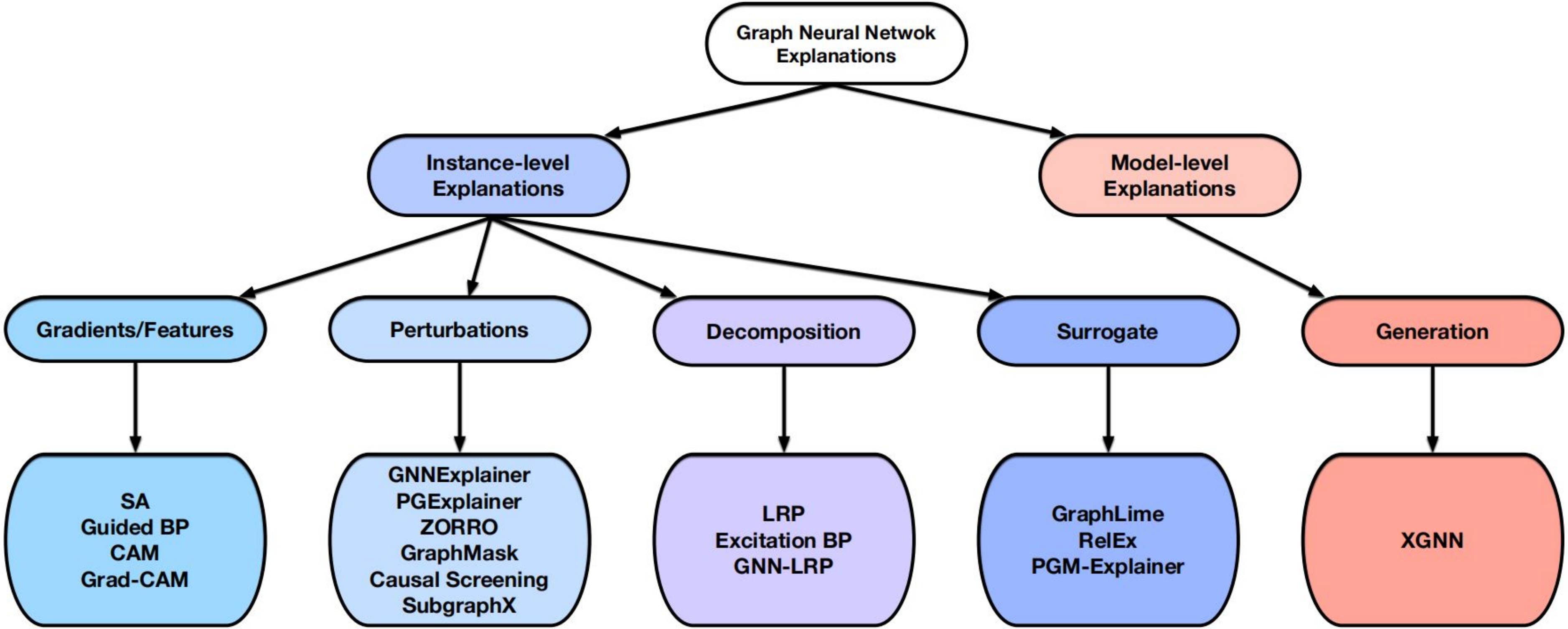


Layer 3: https://ralo93.github.io/xai_gat2/

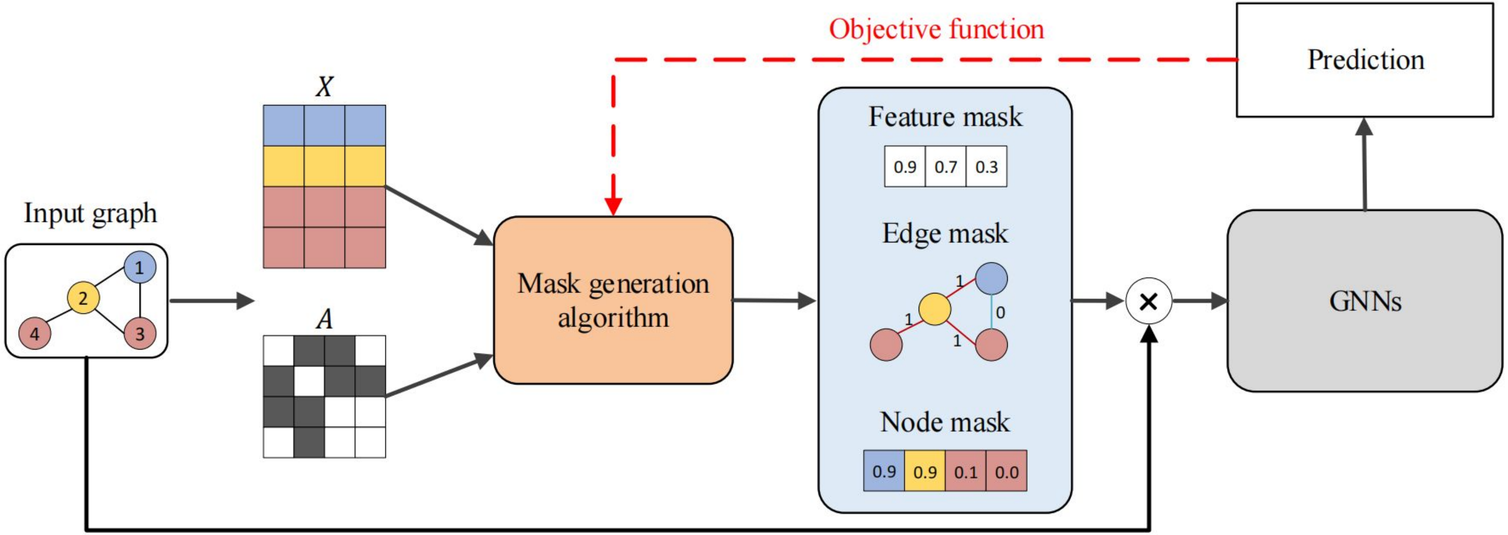
Roman empire dataset: heterophilic graph, 22662 nodes, 18 classes





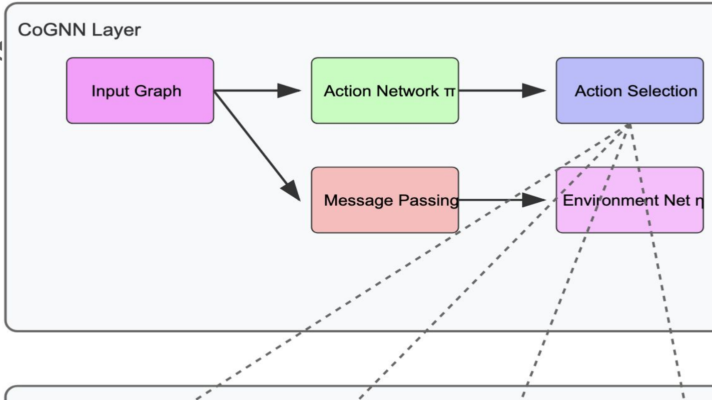

Graph Neural Netwok Explanations
Instance-level Explanations
Model-level Explanations
Gradients/Features
Perturbations
Decomposition
Surrogate
Generation
SA
Guided BP
CAM
Grad-CAM
GNNExplainer
PGExplainer
ZORRO
GraphMask
Causal Screening
SubgraphX
LRP
Excitation BP
GNN-LRP
GraphLime
RelEx
PGM-Explainer
XGNN

mask generation

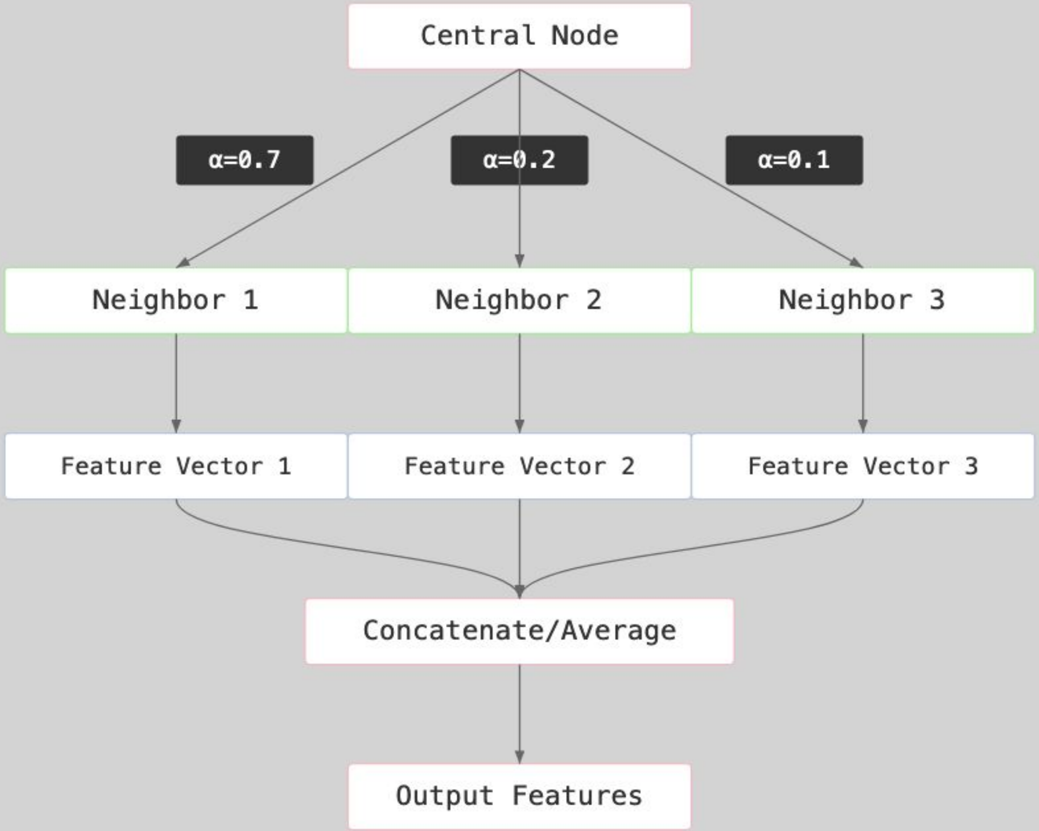


Understanding Graph Datasets

- Multiple node types and/or edge types
- Example: Knowledge graphs with different entity types
- Complex relationships and varying feature spaces
- Roman Empire dataset showcasing diverse entity types



Multi-Head Attention



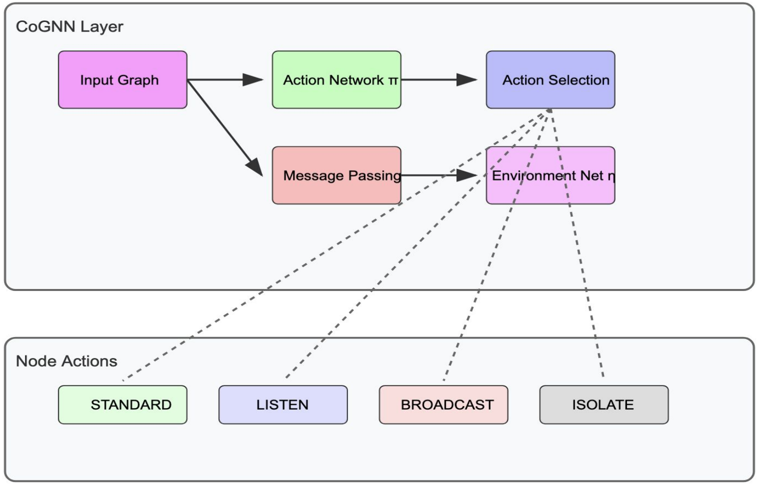
Model Architectures

3. **The Cooperative Graph Neural Networks' (CoGNN) Framework**

Core Concept

→ Nodes as Active Players!

- Dynamic and asynchronous message passing
- Flexible information flow based on node state
- Task-specific topology learning
- Each node can choose its communication strategy



CoGNN - Dual Network Design

<u>Action Network (π)</u>

- Predicts probability distribution over actions
- Shared across layers
- Uses node states to determine optimal actions
- Enables dynamic graph exploration

<u>Environment Network (η)</u>

- Updates node representations
- Processes messages based on chosen actions
- Aggregates information selectively
- Learns task-specific features

CoGNN - Message Passing Mechanism

Two-Stage Process

1. Action Selection
- Each node chooses action using π network
- Actions determine information flow
- Creates dynamic computation graph
- Enables directed message passing

2. State Update
- Environment network η updates nodes
- Updates depend on chosen actions
- Different update rules for different actions
- Selective information aggregation

CoGNN - Advantages and Benefits

Key CoGNN Features

- Mitigates over-smoothing through isolation
- Better handles long-range dependencies
- Task-specific topology learning
- More expressive than standard GNNs
- Efficient runtime complexity

Empirical Results

- State-of-the-art on heterophilic graphs
- Strong performance on long-range tasks
- Effective on both homophilic and heterophilic data
- Robust to increasing network depth