## Contents



**Team members:** 1. Jędrzej Miczke 156068 2. Mateusz Nowicki 156064 3. Krzysztof Skrobała 156039 4. Wojciech Bogacz 156034

# 1 Introduction

This project aims to test methods for predicting a paper category based on its abstract, and the graph structure of citations between papers. The main idea is to reproduce the results from the paper *Semi-Supervised Classification with Graph Convolutional Networks* by Thomas N. Kipf and Max Welling, presented at ICLR 2017, and to compare it with other methdods.

# 2 Related Work

# 3 Dataset description

# 4 Methods Decription

## 4.1 Graph Convolutional Networks (GCN)

Graph Convolutional Networks (GCNs) extend convolution to graph-structured data by iteratively aggregating and transforming features from a node's neighbors using the graph topology.

A common layer-wise propagation rule is:

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right)$$

where:

- $\tilde{A} = A + I$ is the adjacency matrix with self-loops.
- $\tilde{D}$ is the degree matrix of $\tilde{A}$.
- $H^{(l)}$ represents the node features at layer $l$.
- $W^{(l)}$ denotes the learnable weights.

- $\sigma(\cdot)$ is a non-linear activation function.

We can note that when there are no edges, the $\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$ term becomes the identity matrix, and the layer reduces to a standard feedforward layer, which will be useful for a baseline comparison.

### 4.2 Graph Attention Networks (GAT)

Graph Attention Networks (GATs) incorporate an attention mechanism to assign different importance to neighboring nodes when aggregating features.

For a node $i$, attention coefficients are computed as:

$$\alpha_{ij} = \frac{\exp\left(\text{LeakyReLU}\left(\vec{a}^\top[W\vec{h}_i \,\|\, W\vec{h}_j]\right)\right)}{\sum_{k\in\mathcal{N}(i)}\exp\left(\text{LeakyReLU}\left(\vec{a}^\top[W\vec{h}_i \,\|\, W\vec{h}_k]\right)\right)}$$

and the node update is:

$$\vec{h}_i' = \sigma\left(\sum_{j\in\mathcal{N}(i)} \alpha_{ij} W\vec{h}_j\right)$$

where: - $\vec{h}_i$ represents the input node features. - $W$ and $\vec{a}$ are learnable parameters. - $\mathcal{N}(i)$ denotes the neighbors of node $i$. - $\|$ is the concatenation operation. - $\sigma(\cdot)$ is a non-linear activation function.

### 4.3 Graph Sample and Aggregate (GraphSAGE)

GraphSAGE is an inductive graph representation learning method that learns node embeddings by sampling and aggregating features from a fixed-size set of neighboring nodes.

For node $i$ at layer $l$, the neighborhood aggregation is:

$$h_{\mathcal{N}(i)}^{(l)} = \text{AGGREGATE}^{(l)}\left(\{h_j^{(l)} : j\in\mathcal{N}(i)\}\right)$$

and the node update is:

$$h_i^{(l+1)} = \sigma\left(W^{(l)}\cdot[\,h_i^{(l)} \,\|\, h_{\mathcal{N}(i)}^{(l)}]\right)$$

followed by normalization.

where: - $\mathcal{N}(i)$ denotes the **sampled** neighbors of node $i$. - $W^{(l)}$ are the learnable weights. - $\|$ indicates concatenation. - AGGREGATE$(\cdot)$ is a differentiable function (e.g., mean, max-pooling, or LSTM).

### 4.4 Graph Trasnformer

Graph Transformers adapt the self-attention mechanism of Transformers to graph-structured data, enabling nodes to attend to other nodes based on learned attention scores while incorporating graph structure via positional or edge encodings.

For a node $i$, attention is computed as:

$$\text{Attn}(i, j) = \frac{(W_Q h_i)(W_K h_j)^\top}{\sqrt{d}}$$

and the node update is:

$$h'_i = \sum_{j \in \mathcal{V}} \text{softmax}_j \left( \text{Attn}(i, j) + b_{ij} \right) W_V h_j$$

where: * $h_i$ are the node features. * $W_Q, W_K, W_V$ are learnable projection matrices for Query, Key, and Value. * $d$ is the attention dimension (used for scaling). * $b_{ij}$ is a bias term encoding graph structure (e.g., shortest path distance or edge connectivity). * $\mathcal{V}$ is the set of all nodes in the graph.

## 5 Evaluation Strategy

We measure the performance of each model using accuracy on a held-out test set. Additionally, we monitor training and validation loss to assess convergence and potential overfitting.