SPEML Exercise 2
Group 04, Summer 2025

Hedda Fiedler (12402607), Lou Elah Süsslin (12347669)

June 15, 2025

Graph Neural Networks (GNNs) such as Graph Convolutional Networks (GCNs) and Graph Attention Networks (GATs) have become state-of-the-art for node classification tasks (see [1, 2]). However, many real-world graphs—like academic citation networks—are naturally distributed across data silos (e.g., universities, companies, institutions), where direct data sharing is restricted due to privacy or organizational constraints.

We need to pick a specific learning scenario and dataset in this hemisphere to work on. The purpose of this document is to put forward the outline for how we are going to pursue implementing a model as a case study for the chosen topic.

Topic Description

We want to work on a project in the hemisphere of distributed computation, more specifically *Federated Learning on graph data (Topic 2.3.3).* Examples for models are Graph Attention Networks to compute the relation of nodes dynamically using self-attention mechanism, or Graph Convolutional Networks.

Federated Learning (FL) enables collaborative model training without exchanging raw data, making it suitable for settings where data privacy of the different clients working together is important [3].

Problem-solving Approach

According to Zhang et al. [3], there are four categories of how graph data is distributed in FL. One of them is *horizontal intra-graph federated learning*, which we pick as a learning scenario.*Horizontal* refers to the data split: all clients share the same feature and label spaces but hold different nodes (samples). Each client's nodes share the same feature type (e.g., 128-d paper embeddings) and classification task (40 subjects). *Intra-graph* describes how the graph itself is partitioned: the entire dataset forms a single large graph (e.g., the citation network), which is split into subgraphs distributed across clients. Unlike inter-graph FL, clients do not have separate independent graphs but hold portions of one connected graph.

In summary, *horizontal* characterizes the data partition (same features, different nodes), while *intra-graph* captures the graph structure partition (clients hold subgraphs of one large graph).

More concretely, **our learning scenario–and therefore task–follows the approach suggested by Zhang et al. [3], where a single citation network is partitioned across multiple agents, simulating a scenario in which these agents represent different institutions. Each agent holds a disjoint subset of papers, but cross-institutional citations create dependencies that need to be managed without direct data sharing. We select this as our learning scenario.**

We will start with centralized learning by training a Graph Neural Network (GNN) on the entire graph, using all nodes and edges. Next, we transfer to federated learning by partitioning the graph horizontally into subgraphs, each assigned to a different client. Each client trains the model locally on their subgraph, and after local training, federated aggregation methods (see below) are used to combine model weights across clients.

Using the same metrics for validation and testing, we aim to evaluate how the federated graph learning approach compares to the centralized approach on the same classification task, in terms of both effectiveness (e.g., accuracy) and efficiency (e.g., training time). This comparison will also highlight the benefit of preserving data privacy across clients during training.

Method

- Simulate a federated setting with 3–5 clients using simple node partitioning of the arvix graph (see datasets)
- Train local GCN models on each client and aggregate them using Federated Averaging (FedAvg) (e.g. with Flower [1] or FedML[2])
- Evaluate how well a federated model approximates centralized performance (baseline model: centralized GCN on the full graph)

Dataset

We will use the OGBN-Arxiv dataset from the Open Graph Benchmark[3]. It has different sizes; below is an example for size `small`. We may potentially play around with the size during the actual task.

- #Nodes: 169,343
- #Edges: 1,166,243
- Multi-class classification
- Metric: Accuracy
- Simulation of the federated learning setting with multiple clients holding different papers by splitting the graph into n clients.

Metrics

The metrics serve the purpose of comparing the centralized model vs. the federated model in this learning scenario, that is, on the task. Possible metrics according to our sources are:

Effectiveness Metrics	
Classification Accuracy	Standard metric for multi-class node classification tasks on citation graphs (e.g., Cora, Citeseer, Pubmed). Widely used in centralized GNNs (GCN, GAT) and relevant for horizontal intra-graph FGL. [1, 2, 3]
Micro-averaged F1 Score	Applied in multi-label classification tasks (e.g., the protein-protein interaction dataset for inductive learning tasks, where protein nodes can belong to multiple of 121 gene ontology-derived classes, such as “immune response” and “cell proliferation” [2]). Reflects class imbalance better than plain accuracy.
Efficiency Metrics	
Wall-clock Training Time	Total time required for training until convergence. Reported in GCN and GAT papers to compare practical efficiency. Especially critical in FL where communication adds latency. [1, 2, 3]
Convergence Speed	*As Accuracy or Loss over Rounds.* Indicates how quickly the model stabilizes. In FL, convergence can be slowed by non-IID data or graph partitioning. GCN and FGL works monitor validation metrics for early stopping. [1, 2, 3]
Communication Overhead	Measures data exchanged between server and clients per round. Cited in FGL literature as a core bottleneck due to large model sizes and frequent updates. [3]

Table 1: Metrics for evaluating centralized and federated models in horizontal intra-graph learning tasks, grouped by evaluation category.

[1] https://flower.ai/
[2] https://fedml.ai/home
[3] https://ogb.stanford.edu/docs/nodeprop/ogbn-arxiv

References

[1] Kipf, T. N. and Welling, M. (2017). Semi-supervised classification with graph convolutional networks.

[2] Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. (2018). Graph attention networks.

[3] Zhang, H., Shen, T., Wu, F., Yin, M., Yang, H., and Wu, C. (2021). Federated graph learning – a position paper.