A Comparative Study of DeepWalk and GNN Models in Graph Embedding

Hatim Mrabet, Abdennacer Badaoui

April 7, 2025

1 Introduction

Graph Neural Networks (GNNs) have recently become state-of-the-art for learning on graph-structured data, excelling in tasks like node classification and graph clustering. Classical approaches such as random walks, though simpler, remain effective for capturing graph topology and generating node embeddings.

In this project, we compare the performance of GNN-based models (a basic GNN with MLP and a Graph Attention Network) with a random walk-based method on benchmark datasets: **Cora**, **Citeseer**, and **Amazon Computers**.

We evaluate how well each model performs on node classification, providing insight into their ability to learn class-discriminative representations. Additionally, we visualize the learned embeddings using *t-SNE* to qualitatively assess how effectively each model separates the classes in the embedding space.

This comparative study highlights the trade-offs between traditional and neural methods for graph representation learning.

For the implementation details, we have developed all the components presented in this project from scratch — including architectures such as GNN and GAT, as well as DeepWalk and the random walk generation — without relying on libraries like PyTorch Geometric. The only exception is Word2Vec, for which we used an existing implementation from a library.

Hatim Mrabet primarily concentrated on experiments involving GNN and GAT models, while Abdennacer Badoaui focused on recreating the DeepWalk framework from scratch.

2 Code Implementation

The full implementation of models, along with the experiments and evaluations, is available in this **Github Repository**

3 Theoretical Overview

3.1 Graph Neural Networks (GNNs)

Graph Neural Networks (GNNs) are designed to learn node representations by aggregating information from neighbors, capturing the graph's structure.

Let $G = (V, E)$ be an undirected graph, and $h_v^{(0)} \in \mathbb{R}^d$ the initial feature of node $v \in V$. A GNN layer updates embeddings as:

$$h_v^{(k)} = \mathrm{MLP}^{(k)} \left(\mathrm{AGG} \left(\left\{ h_u^{(k-1)} : u \in \mathcal{N}(v) \cup \{v\} \right\} \right) \right),$$

where $\mathcal{N}(v)$ is the neighborhood of v, AGG is an aggregation function (e.g., mean or sum), and MLP is a learnable multi-layer perceptron. This approach, often called **GNN with MLP**, captures non-linear patterns while remaining simple and efficient.

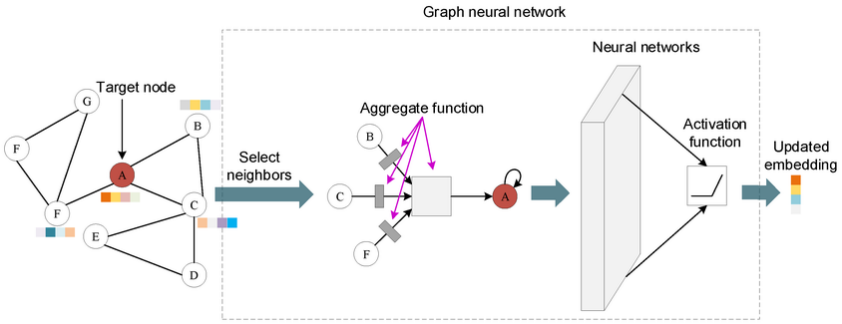

Figure 1: Graph Neural Networks Architecture

3.2 Graph Attention Networks (GATs)

Graph Attention Networks (GATs) enhance GNNs by assigning different attention weights to neighbors, allowing nodes to focus on the most informative ones.

For nodes v and u, the attention coefficient is computed as:

$$\alpha_{vu} = \frac{\exp\left(\text{LeakyReLU}\left(a^\top [Wh_v \,\|\, Wh_u]\right)\right)}{\sum_{k \in \mathcal{N}(v)} \exp\left(\text{LeakyReLU}\left(a^\top [Wh_v \,\|\, Wh_k]\right)\right)},$$

with W as a learnable weight matrix, a an attention vector, and $\|$ denoting concatenation. The updated embedding is:

$$h'_v = \sigma\left(\sum_{u \in \mathcal{N}(v)} \alpha_{vu} W h_u\right).$$

GATs allow more expressive message passing by learning to weigh each neighbor's contribution dynamically.

3.3 DeepWalk: Random Walk-Based Representation Learning

DeepWalk is a classical, unsupervised method for learning node representations by leveraging random walks combined with language modeling techniques. The core idea is to generate sequences of nodes through truncated random walks and treat them analogously to sentences in natural language.

Given a graph $G = (V, E)$, DeepWalk performs multiple random walks of fixed length l starting from each node. Each walk produces a sequence $(v_0, v_1, \ldots, v_l)$ where each node v_{i+1} is sampled uniformly from $\mathcal{N}(v_i)$, the neighbors of v_i.

These sequences are then fed into the Skip-Gram model to maximize the likelihood of predicting a node's context:

$$\max_{\Phi} \sum_{v \in V} \sum_{u \in \mathcal{C}(v)} \log \Pr(u | \Phi(v)),$$

where $\Phi : V \to \mathbb{R}^d$ maps nodes to embeddings, and $\mathcal{C}(v)$ denotes the context window around node v in the walk.

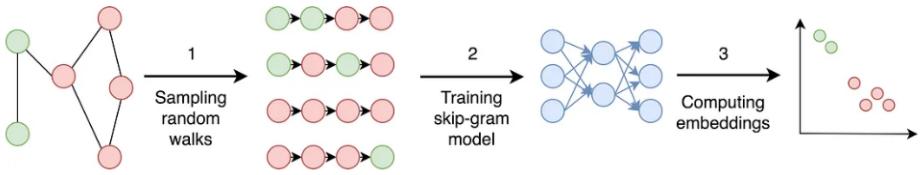

Figure 2: DeepWalk: Random walks and Skip-Gram-based embedding learning

DeepWalk effectively captures structural similarity by placing nodes that frequently co-occur in walks close in the embedding space. In our project, we use it as a baseline to evaluate how well such shallow, unsupervised embeddings perform compared to supervised GNN-based approaches on node classification tasks.

4 Experiments and Results

4.1 Experimental Setup

To evaluate the performance of different graph-based models on node prediction tasks, we conduct experiments on three benchmark datasets: Cora (2,708 nodes, 5,429 edges), Citeseer (3,327 nodes, 4,732 edges), and Amazon Computers (13,752 nodes, 245,861 edges). These datasets represent citation and co-purchase networks with varying scales and structural properties. We compare three methods: Graph Neural Networks (GNN), Graph Attention Networks (GAT), and DeepWalk. For DeepWalk, we generate 100 random walks of length 10 per node and learn node embeddings of dimension 128. The embeddings from each model are then used to train a classifier for node label prediction, allowing us to assess the accuracy and generalization ability of each method across different graph types.

4.2 Results

Primary results

To first validate our implementation of the DeepWalk approach, we tested it on a French Webgraph dataset from Kaggle, which contains hyperlinks between various French websites. We generated a t-SNE visualization to inspect the clustering of the learned embeddings, aiming to observe whether links on similar topics are positioned close to each other in the embedding space.

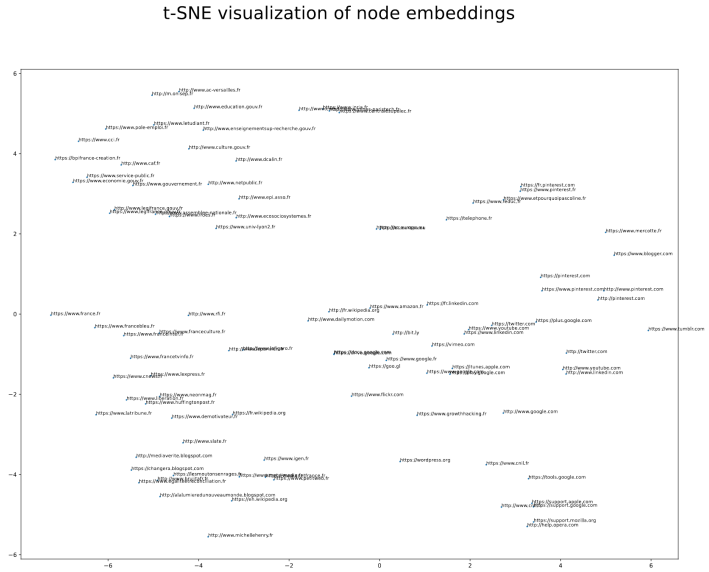

Figure 3: t-SNE visualization of the DeepWalk embeddings on the French Webgraph dataset.

If we focus on specific regions of the visualization, we can clearly observe that links related to the same topic are grouped together. For example, in the two highlighted clusters, the first corresponds to news platforms, while the second includes websites related to research and education.

Figure 4: News platforms cluster

Figure 5: Research and education cluster

Results

The t-SNE visualizations (Fig. 6) of the node embeddings generated by the Graph Neural Network (GNN) model for Citeseer, Cora, and Amazon-Computers show a noticeable degree of class separability. In Citeseer and Cora, nodes belonging to the same class tend to form identifiable clusters, though with some overlap and fragmentation, particularly in Citeseer. In the Amazon-Computers dataset, while large regions of the same class emerge, the boundaries between some classes remain unclear, suggesting that GNN captures structural features but struggles with sharper decision boundaries, especially in larger, denser graphs.

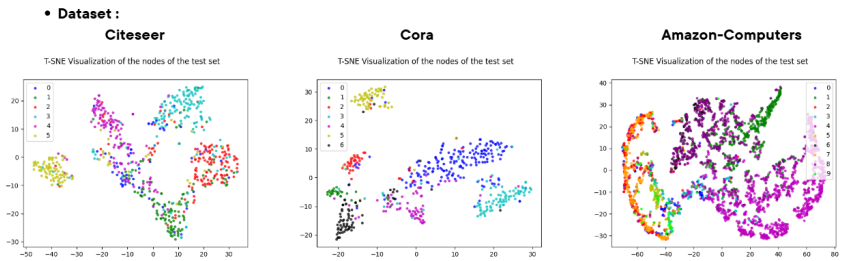

Figure 6: T-SNE visualization of GNN features across different datasets

Graph Attention Networks (GAT) significantly improve the quality of the learned embeddings, as evident from the tighter and more distinct clusters in the t-SNE plots (Fig. 7). In the Citeseer and Cora datasets, GAT embeddings show better separation between classes, reducing the inter-class overlap observed with GNN. The improvement is especially noticeable in the Amazon-Computers dataset, where GAT yields clearer and more compact clusters, reflecting its ability to assign different attention weights to neighbors and focus on the most informative connections. This highlights GAT's advantage in learning more discriminative node representations, particularly in complex or noisy graph structures.

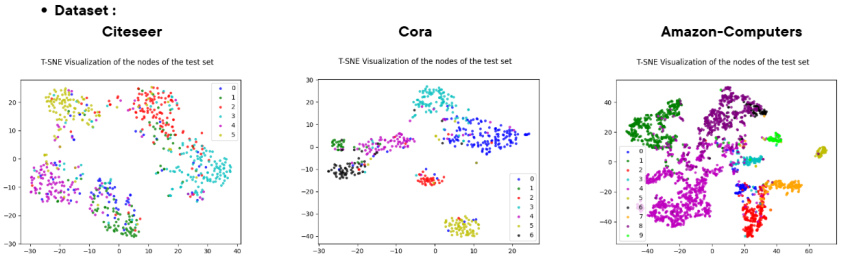

Figure 7: T-SNE visualization of GAT features across different datasets

On the DeepWalk side, Cora exhibits well-separated clusters, indicating that DeepWalk effectively captures the graph structure, resulting in clear class distinctions (Fig. 8). Amazon-Computers also displays identifiable clusters, though with more overlap than Cora, suggesting that while DeepWalk captures broader community structures, it may struggle with finer class boundaries. In contrast, Citeseer shows a more scattered distribution with less distinct clustering and higher overlap between classes. This suggests that DeepWalk has difficulty learning meaningful representations for Citeseer, potentially due to higher class similarity or a more complex and sparse graph structure. Overall, DeepWalk performs best on Cora, followed by Amazon-Computers, and faces the most challenges with Citeseer.

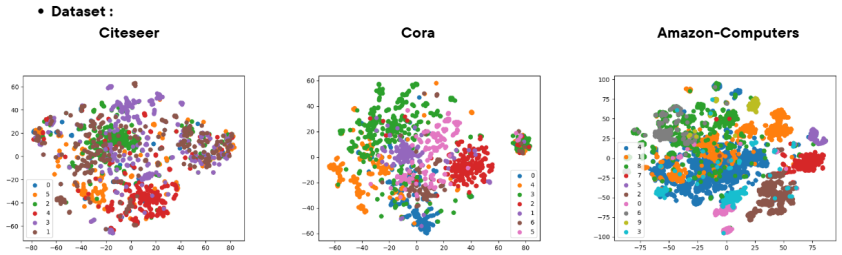

Figure 8: T-SNE visualization of DeppWalk features across different datasets

5 Conclusion

The table 1 compares node classification accuracy across three datasets using GNN, DeepWalk, and GAT. On Cora, GNN (85.61) slightly outperforms both DeepWalk (85.06) and GAT (84.32), suggesting that in citation networks with strong homophily, neighborhood aggregation enhances predictive performance, while DeepWalk's structural embeddings remain competitive. For Citeseer, GNN (69.97) and GAT (69.82) significantly outperform DeepWalk (59.60), likely due to the dataset's lower connectivity and higher heterophily, which reduce the effectiveness of DeepWalk's random-walk-based embeddings. In contrast, on Amazon-Computers, DeepWalk (88.57) outperforms GAT (87.06) and significantly surpasses GNN (63.47), indicating that in large-scale, loosely structured graphs, DeepWalk captures useful structural patterns more effectively than message-passing methods, which may suffer from over-smoothing. Overall, GNNs perform best in homophilous graphs like Cora and Citeseer, while DeepWalk proves more effective in large, less structured networks such as Amazon-Computers.

Dataset	**GNN**	**DeepWalk**	**GAT**
Cora	**85.61**	85.06	84.32
Citeseer	**69.97**	59.60	69.82
Amazon Computers	63.47	**88.57**	87.06

Table 1: Comparative Results: Node prediction accuracy