Graph Attention Networks: Reproducibility Study

Sina Hassanpouryousefi (GTid: shassanp3)

Team 32

Georgia Tech

Project Overview and Motivation

- Objective: Reproduce and extend the GAT architecture from Velicković et al. (ICLR 2018).
- Why GATs?
  - Graphs are ubiquitous (e.g., citation networks, social networks, molecular graphs).
  - GATs use masked self-attention to weigh neighbor importance, unlike spectral methods (e.g., GCNs).
  - State-of-the-art performance on node classification tasks (Cora, Citeseer, Pubmed).
- Scope: Implement GAT and baselines (GCN, GraphSAGE, ChebNet, etc.) using PyTorch Geometric, test on Cora dataset, and validate hypotheses.

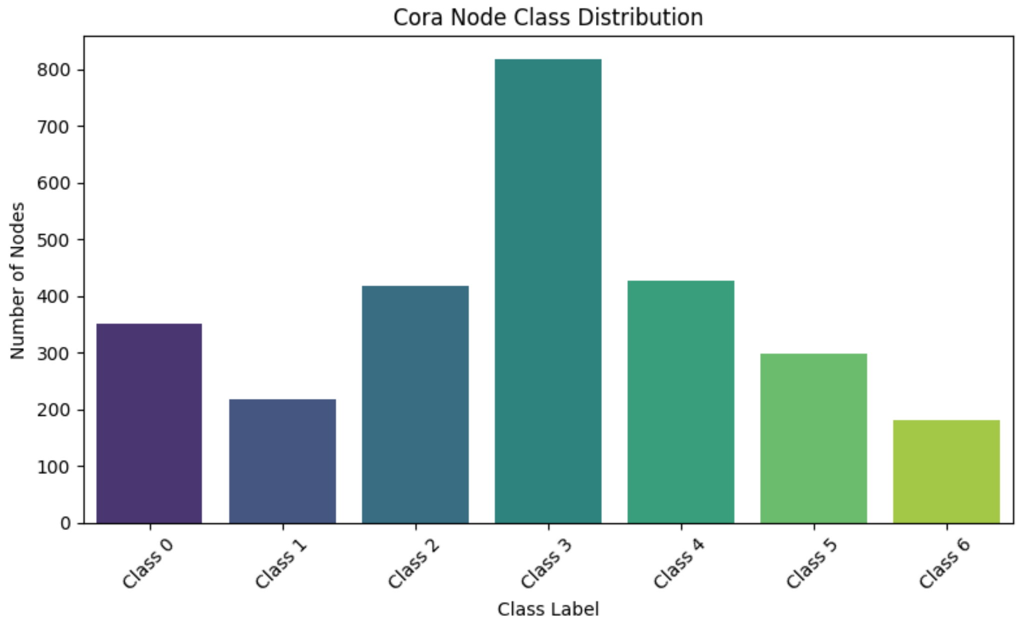
Hypotheses and Experiments

- Hypotheses Tested (from GAT paper):
  - H1: GAT outperforms GCN and ChebNet on node classification (Cora dataset).
  - H2: Attention mechanism prioritizes semantically relevant neighbors.
  - H3: Multi-head attention improves stability and performance.

- Experiments:
  - Train GAT, GCN, ChebNet, GraphSAGE, etc., on Cora (140 train, 500 val, 1000 test nodes).
  - Visualize attention coefficients using t-SNE.

Methodology and Dataset

- Dataset: Cora (Planetoid benchmark)
  - Nodes: 2708, Edges: 10556 (undirected), Features: 1433 per node, Classes: 7.
  - Moderately imbalanced (e.g., Class 3: 818 nodes, Class 6: 180 nodes).
  - Preprocessing: Row-normalized features, used PyTorch Geometric's train/val/test masks.
- Models Implemented:
  - GAT (gat.py): 2 layers, 8 heads (Layer 1), ELU, dropou (0.6).
  - Baselines: GCN, ChebNet, GraphSAGE (Mean, Pooling), SemiEmb, GatedGCN.
- Framework: PyTorch Geometric, train.py for unified training/evaluation.



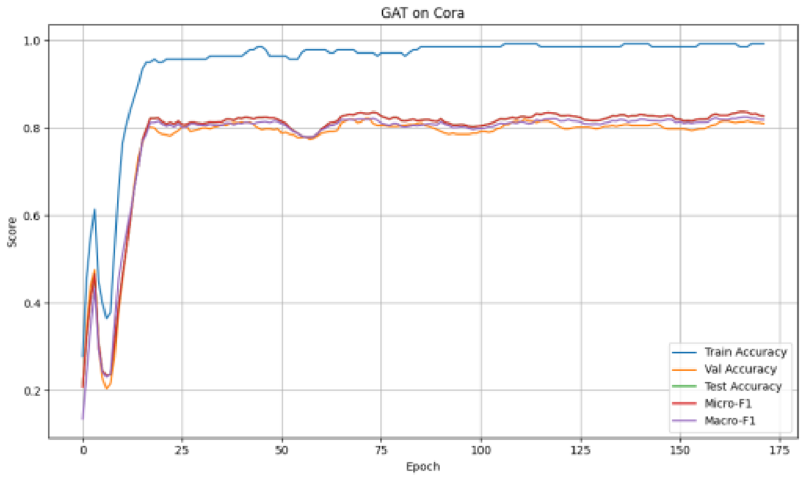
Training Setup

- Hardware: Intel Core i9-12900K CPU, 64 GB RAM, Windows 10 (no GPU).
- Framework: PyTorch 2.6.0 (CPU), PyTorch Geometric.
- Training Details:
  - Loss: Negative Log-Likelihood (NLLoss).
  - Optimizer: Adam (lr=0.005, weight decay: 5e-4 on Layer 1, 0 on Layer 2).
  - Regularization: Dropout (0.6), early stopping (patience=100 epochs).
  - Metrics: Accuracy, Micro-F1, Macro-F1.
- Runtime: ~0.15s/epoch, ~ 250 epochs (avg.), 50 trials total.

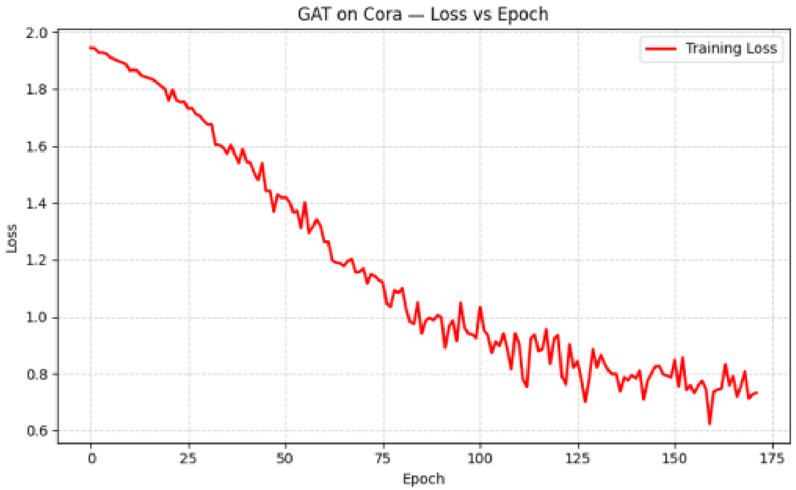
Results and Evaluation

- Test Accuracy (Cora):
  - GAT: 83.3% (Original: 83.0%)
  - GCN: 82.0% (Original: 81.5%)
  - ChebNet: 81.8% (Original: 81.2%)
  - GraphSAGE: 79.7%,
  - SemiEmb: 58.6%,
  - GatedGCN: 73.2%.

Model	Ours (Reproduced)	Original Paper
GAT	83.3	83.0
GCN	82.0	81.5
GraphSAGE	79.7	—
GatedGCN	73.2	—
ChebNet	81.8	81.2
SemiEmb	58.6	59.6
GraphSAGE-Mean	79.7	—
GraphSAGE-Pooling	77.2	—



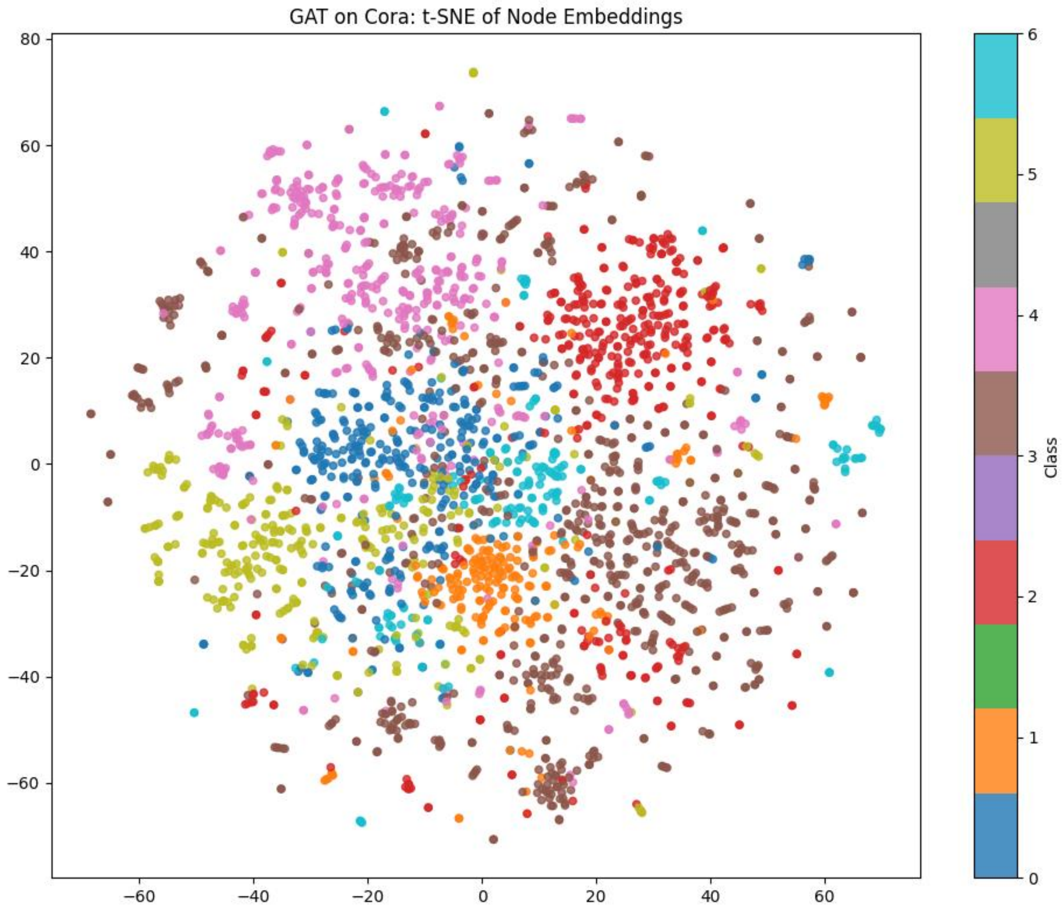
(a) Accuracy & F1 over Epochs



(b) Loss vs. Epoch

Attention Visualization and Insights

- Experiment: Visualized attention coefficients using t-SNE projections (H2).
- Findings: High-attention links often connect same-class nodes, confirming attention's semantic focus.
- Challenges: Extracting coefficients required custom hooks in PyTorch Geometric.



Discussion and Reproducibility

- Reproducibility: GAT is highly reproducible (83.3% vs. 83.0% original).
- Facilitators:
    - Clear GAT paper documentation.
    - PyTorch Geometric's Cora dataset access.
    - Modular train.py pipeline.
    - LLM assistance for code and metrics.
- Challenges:
    - Limited dataset access (only Cora).
    - Attention visualization required custom tools.
    - Dropout/weight decay tuning was critical.

Recommendations and Future Work

- For Paper Authors:
  - Share full configuration scripts (e.g., optimizer, seeds).
  - Provide utilities for attention visualization.
  - Include preprocessing for inductive datasets (e.g., PPI).
- For Reproducers:
  - Use PyTorch Geometric’s Planetoid datasets with clear preprocessing.
  - Specify LLM prompts for masking/regularization.
  - Verify edge symmetry, test multiple seeds.
- Future Work:
  - Test on Citeseer, Pubmed, PPI.
  - Integrate edge features into attention.
  - Extend to graph-level classification.

Georgia Tech

Conclusion

- Key Takeaways:
  - Successfully reproduced GAT (83.3% accuracy on Cora, vs. 83.0% original).
  - Validated hypotheses: GAT outperforms baselines, attention is critical, multi-head stabilizes.
  - Built a modular, extensible pipeline for GNN benchmarking.
- Impact: Provides a reproducible framework for GNN research, with insights into attention's role.
- Closing: Questions? Explore the GitHub repo for code and details.