# CSE 6389 Special Topics in Advanced Multimedia, Graphics, & Image

# GNN for AD classification using brain connectivity Fall 2025

# Project 2

**Instructor:**
Dr. Dajiang Zhu

**Submitted By:**
Srinivasa Sai Abhijit Challapalli.

1002059486

## 1. Abstract

This project implements a compact Graph Convolutional Network (GCN) for binary classification of Alzheimer's Disease (AD) versus Cognitively Normal (CN) subjects using 150×150 brain connectivity matrices. Structural connectivity (SC) is treated as the message passing topology (graph adjacency) after GCN symmetric normalization, while functional connectivity (FC) is used as node features after optional Fisher z-transform and standardization. The primary emphasis is on a sound, reproducible pipeline preprocessing, graph construction, and stratified k-fold evaluation rather than raw accuracy on this small dataset (N=20).

## 2. Dataset

• Subjects: 20 total (10 AD, 10 CN).

• Per subject: two 150×150 matrices in text format: FunctionalConnectivity.txt (FC) and StructuralConnectivity.txt (SC).

• Directory layout:

```
project_root/
 ├── AD1/
|    ├── FunctionalConnectivity.txt
|   └── StructuralConnectivity.txt
 ├── AD2/
|    ├── FunctionalConnectivity.txt
|   └── StructuralConnectivity.txt
 ├── ...
 ├── CN1/
|    ├── FunctionalConnectivity.txt
|   └── StructuralConnectivity.txt
└── CN10/
    ├── FunctionalConnectivity.txt
   └── StructuralConnectivity.txt
```

## 3. Preprocessing & Graph Construction

Structural Connectivity (SC) → Adjacency

1) Symmetrize and zero the diagonal.
2) Apply log1p to compress heavy-tailed weights (optional unit scaling to [0, 1]).
3) Add self-loops and compute GCN normalization: ^A = D^{-1/2}(A + I)D^{-1/2}.

Functional Connectivity (FC) → Node Features

1) Optionally apply Fisher z-transform (arctanh of clipped correlations).
2) Standardize either globally per subject or row-wise per node.
3) Use each row (length 150) as the feature vector for the corresponding node.

## 4. Model Architecture

I reviewed recent works combining FC & SC with GNNs and distilled the following takeaways for a small dataset (N=20):

Joint-GCN (DTI + rs-fMRI, joint graph): builds separate SC and FC graphs and adds learnable inter-network links between matching ROIs; shared encoders and fusion typically outperform single-modality GCNs.
Takeaway: light cross-modal coupling helps but adds complexity.

MMTGCN (Mutual Multi-Scale Triplet GCN): multi-scale graphs and triplet interactions capture higher-order relationships; shows robust gains but is heavier to implement.
Takeaway: multi-scale and modality-aware branches can improve robustness, but may be overfit with very small N.

GCNNs for AD spectrum: two-layer GCNs with GCN normalization and graph-level pooling are strong baselines on connectome data.
Takeaway: a compact two-layer GCN is a solid starting point.

SC ↔ FC relation with GCN encoders: SC provides a stable anatomical backbone that constrains FC; using SC as the message-passing graph and injecting FC as features is principled.
Takeaway: use SC for topology and FC as features.

**What I implemented**

Given the dataset size, I adopted the simplest defensible design: single-branch GCN over SC with FC as node features. A dual-branch fusion or a cross-modal gate is left as future work/ablation, consistent with Joint-GCN’s motivation.

I implemented a two-layer GCN with batch normalization, dropout, and a small MLP head. SC provides the graph topology for message passing, while FC provides node features. Global mean pooling aggregates node embeddings into a graph-level embedding per subject.

**Final architecture (per subject):**
- Input: A ∈ R{150×150} (from SC), X ∈ R {150×150} (from FC).
- Linear(150 → 64) → ReLU → Dropout(0.5)
- GCN layer (64 → 64) with pre-normalized ^A → ReLU → Dropout(0.5)
- GCN layer (64 → 64) → Global mean pool over nodes → 64-d vector
- MLP head: Linear(64 → 64) → ReLU → Dropout(0.5) → Linear(64 → 2)

## 5. Training Configuration

• Loss: CrossEntropyLoss

• Optimizer: AdamW (typical settings: lr=1e-3, weight_decay=1e-6)

• Regularization: Dropout and gradient clipping (max-norm 2.0)

• Early stopping: patience based on validation ROC-AUC

• Cross-validation: StratifiedKFold (default 5 folds)

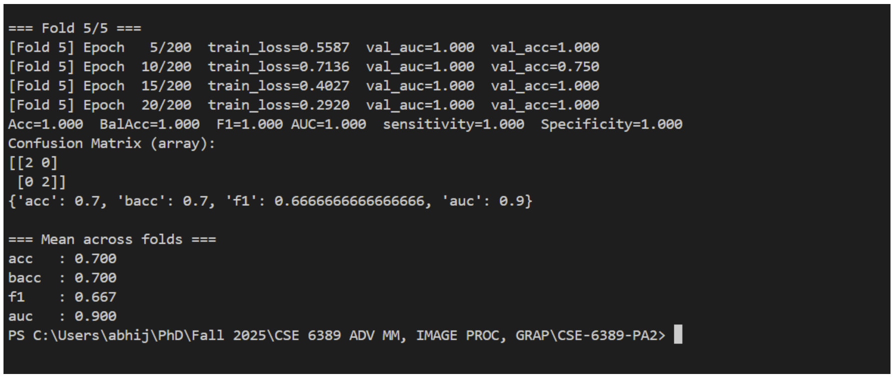
• Metrics: Accuracy, Balanced Accuracy, F1, ROC-AUC, Recall, Precision

• Artifacts: Confusion matrix heatmap per fold saved as cm_fold_{k}.png

## 6. Evaluation and Results

The script prints per-fold metrics and saves a confusion matrix image per fold (cm_fold_k.png) where available. On this small dataset, mean performance is reported across folds. Results will vary based on random seeds and hyperparameters.

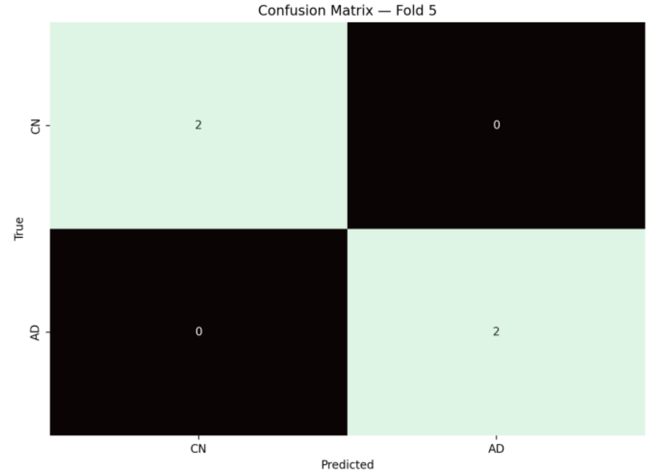
Mean across folds
acc   : 0.700
bacc : 0.700
f1   : 0.667
auc  : 0.900
Fold 5 training results:



```
=== Fold 5/5 ===
[Fold 5] Epoch   5/200  train_loss=0.5587  val_auc=1.000  val_acc=1.000
[Fold 5] Epoch  10/200  train_loss=0.7136  val_auc=1.000  val_acc=0.750
[Fold 5] Epoch  15/200  train_loss=0.4027  val_auc=1.000  val_acc=1.000
[Fold 5] Epoch  20/200  train_loss=0.2920  val_auc=1.000  val_acc=1.000
Acc=1.000  BalAcc=1.000  F1=1.000 AUC=1.000  sensitivity=1.000  Specificity=1.000
Confusion Matrix (array):
[[2 0]
 [0 2]]
{'acc': 0.7, 'bacc': 0.7, 'f1': 0.6666666666666666, 'auc': 0.9}

=== Mean across folds ===
acc   : 0.700
bacc  : 0.700
f1    : 0.667
auc   : 0.900
PS C:\Users\abhij\PhD\Fall 2025\CSE 6389 ADV MM, IMAGE PROC, GRAP\CSE-6389-PA2>
```

Fold 5 confusion matrix:



## 7. Ablation Study

- FC role: Compare FC as adjacency (|z|, sparsified) versus FC as features (baseline).
- Normalization choices: With/without Fisher z-transform; global vs row-wise standardization.
- Hidden size / depth: Vary hidden dimension and number of GCN layers (1–3).

## 8. Reproducibility

We fix seeds for NumPy and PyTorch and set deterministic cuDNN flags when possible. Given the data size, even small changes to initialization may affect fold-level outcomes, so multiple runs are recommended.

## 9. Repository Structure

```
project_root/
 ├── AD*/ CN*/              # subject folders with .txt matrices
 ├── train.py              # training + evaluation entry point
 ├── SimpleGCN.py            # Sample GCN code provided
 ├── requirements.txt
 ├── Project2_report.pdf       # document for the project (optional)
 ├── cm_fold_1.png           # generated confusion matrices

 ├── cm_fold_2.png

 ├── cm_fold_3.png

 ├── cm_fold_4.png
 ├── cm_fold_4.png
└── README.md
```

The rest of the plots and the code are available in GitHub:

https://github.com/AbhijitChallapalli/CSE-6389-PA2

## 10. References

Kipf, T. N., & Welling, M. (2017, April). Semi-supervised classification with graph convolutional networks. International Conference on Learning Representations (ICLR). https://arxiv.org/abs/1609.02907

Li, Y., Wei, Q., Adeli, E., Pohl, K. M., & Zhao, Q. (2022). Joint graph convolution for analyzing brain structural and functional connectome. In Medical Image Computing and Computer Assisted Intervention – MICCAI 2022 (Lecture Notes in Computer Science). Springer. https://doi.org/10.1007/978-3-031-16431-6_22

Yao, D., Liu, M., Lin, C., Li, H., Zhang, J., & Shen, D. (2021). A mutual multi-scale triplet graph convolutional network for classification of brain disorders using functional or structural connectivity. IEEE Transactions on Medical Imaging, 40(4), 1279–1289. https://doi.org/10.1109/TMI.2021.3051604

Dsouza, N. S., Nebel, M. B., Crocetti, D., Robinson, J., Mostofsky, S., & Venkataraman, A. (2021, July). M-GCN: A multimodal graph convolutional network to integrate functional and structural connectomics data to predict multidimensional phenotypic characterizations. In Proceedings of the 4th Conference on Medical Imaging with Deep Learning (MIDL) (pp. 119–130). PMLR. https://proceedings.mlr.press/v143/dsouza21a.html

Li, Y., Shafipour, R., Mateos, G., & Zhang, Z. (2019, November). Mapping brain structural connectivities to functional networks via graph encoder–decoder with interpretable latent embeddings. In 2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP). https://www.hajim.rochester.edu/ece/sites/gmateos/pubs/brain/SCFC_GLOBALSIP19.pdf

Liu, J., Ma, G., Jiang, F., Lu, C.-T., Yu, P. S., & Ragin, A. B. (2019). Community-preserving graph convolutions for structural and functional joint embedding of brain networks. arXiv Preprint. https://arxiv.org/abs/1911.03583