Multi-Modal Graph Construction and Classification for Autism Spectrum Disorder using Stacked GCNs

Ravi Shankar Reddy Bedadhala[1], Vaishnavi Duggirala[2], Ram Narayana Malisetti[3] and Sateesh Kumar Reddy C[4]
{ravishankarreddybedadhala@gmail.com[1] duggirala.vaishnavi2003@gmail.com[2]
ramnarayana194@gmail.com[3], satishreddic@gmail.com[4] }

Department of Advanced Computer Science and Engineering, Vignan's Foundation for Science, Technology & Research (Deemed to be University), Guntur, Andhra Pradesh, India[1, 2, 3, 4]

Abstract. Multimodal data are essential to improve the precision and effectiveness of diagnosing neurological conditions such as Autism Spectrum Disorder (ASD). This study leverages functional magnetic resonance imaging (fMRI) time series data, processed using Harvard-Oxford (HO) and Automated Anatomical Labeling (AAL) atlases, alongside non imaging demographic features (e.g., gender, site), to improve ASD classification in complex clinical settings. We propose a multimodal framework that begins with recursive feature elimination (RFE), tailored for ASD tasks involving 1D fMRI time series, to isolate the most significant features contributing to class distinction and reduce dimensionality while preserving critical temporal patterns. The selected attributes are used to formulate a functional connectivity graph of the whole brain. This graph is processed through a hybrid architecture: a graph convolutional network (GCN) with residual connections to mitigate information loss and stabilize gradients, and a Stacked Graph Convolutional Network (GCN) that leverages random walk embeddings to capture higher-order structural relationships. The outputs of the DeepGCN with residual connections and the Stacked GCN are combined and subsequently fed into a Multi-Layer Perceptron (MLP) for binary classification of ASD versus typical controls. To further enhance the model, we introduce an adaptive weighted edge pruning mechanism during training, dynamically adjusting edge weights to optimize graph sparsity while preserving essential connectivity, and an EdgeDrop strategy to randomly sparsify node accuracy connections, reducing overfitting and over smoothing in DeepGCN training. Evaluated using nested 10-fold cross-validation, the adopted methodology results in an ACC of 81. 29% and an AUC of 0.85 for the identification of ASD, outperforming the baseline models and demonstrating the efficacy of integrating feature selection with graph-based learning for accurate and efficient diagnosis.

Keywords: Autism spectrum disorder, multimodal data, graph convolutional network, stacked graph convolutional network, random walk embeddings, edge pruning.

1 Introduction

Autism Spectrum Disorder (ASD) is a complex neurodevelopmental disorder that affects individuals' perceptions and interactions with the world, manifesting as differences in social interaction, communication, and behavior. Those with ASD experience significant difficulties in interpreting nonverbal cues, such as facial expressions, intonation, and body language, and struggle with turn-taking in conversations and adapting to changes in daily routines. The severity of these challenges varies considerably across individuals (Ma et al., 2024; Parisot et al., 2017). Early identification of ASD is critical, as it enables tailored interventions that may

ICITSM-Part II 2025, April 28-29, Tiruchengode, India

DOI 10.4108/eai.28-4-2025.2357985

improve language development, social relationships, and overall quality of life (Zwaigenbaum et al., 2015; Masi et al., 2017). However, diagnosing ASD remains a major challenge due to its complex nature, as traditional diagnostic methods rely heavily on subjective assessments, such as behavior observation and caregiver interviews, which can vary in reliability and interpretation (Reddy et al., 2022; Falkmer et al., 2013; Loth et al., 2016; Hyde et al., 2019).

These conventional diagnostic approaches are limited in terms of precision and reliability as they tend to be influenced by factors such as the experience of the observer, cultural dependency, or the likelihood of an individual masking symptoms (Parisot et al., 2018). Furthermore, these evaluations are time-consuming, potentially postponing much-needed intervention, particularly in underserved areas or for families with limited specialist access. This emphasizes the need for new, objective, and efficient diagnostics (e.g., based on technology, biomarkers, and data-driven insights) that would make it easier to detect ASD early, remove the bias of human components, and facilitate the screening process (Ktena et al., 2018). Further developments in these fields have the potential to transform ASD detection and treatment, providing hope for more accessible and timely support for affected individuals and their families (Yousefian et al., 2022).

However, despite stellar advances in these techniques, a major limitation remains: many classic deep learning architectures function within a Euclidean data structure, and are incapable of capturing the complex topological structure of brain connectivity networks (Li et al., 2021). In response to this limitation, graph-based representations have emerged as a more intuitive framework to model functional and structural networks of the brain, where the nodes index the brain areas and the edges represent the relationships among them (Yang et al., 2021). These graph representations model the non-linear hierarchical characteristics of neural interactions more effectively than other representations. However, both normal CNNs and RNNs struggle to directly process such non-Euclidean data, requiring specialized methods such as Graph Neural Networks (GNNs) to exploit these complex relations (Khodatars et al., 2021).

2 Literature Review

Graph convolutional networks (GCNs) were introduced as a breakthrough for graph-based learning and have been widely used in applications such as neuroimaging and functional connectivity analysis (Zhou et al., 2024). GCNs have been applied to population-based disease prediction based on fMRI data, where demographic information was integrated with the graph structure to enhance classification results (Bandara et al., 2024). The integration of non-imaging information with neuroimaging-based deep learning models represented a step forward in improving diagnosis (Atlam et al., 2025).

Further developments in graph convolutional models for brain connectivity networks analysis have demonstrated significant improvements in diagnosing brain diseases (Cui et al., 2023). Strategies for feature selection, such as Recursive Feature Elimination (RFE), have been used to select the most informative functional connectivity patterns, emphasizing how dimensionality reduction can increase the interpretability and performance of models (Shao et al., 2023).

Deep learning-based ASD diagnosis methods using fMRI data have shown high classification accuracy, demonstrating the effectiveness of unsupervised learning algorithms in neuroimage

analysis (Nafisah et al., 2025). Additionally, advancements in domain adaptation techniques have helped to reduce variability across multisite neuroimaging datasets, improving the generalization of models across diverse populations (Wang et al., 2024).

A hybrid approach combining GCN with multi-layer perceptron (MLP) models for ASD classification has shown improved classification performance (Gautam et al., 2023). Stacked GCN architectures for ASD classification, which leverage hierarchical feature extraction, have enhanced predictive accuracy, outperforming conventional methods (Rubio-Martín et al., 2024).

Finally, innovative frameworks for graph learning that integrate self-attention mechanisms have been proposed to capture complex brain connectivity patterns, offering new benchmarks for functional connectivity analysis (Shao et al., 2023). Their approach outperformed traditional GCNs, setting new benchmarks in functional connectivity analysis.

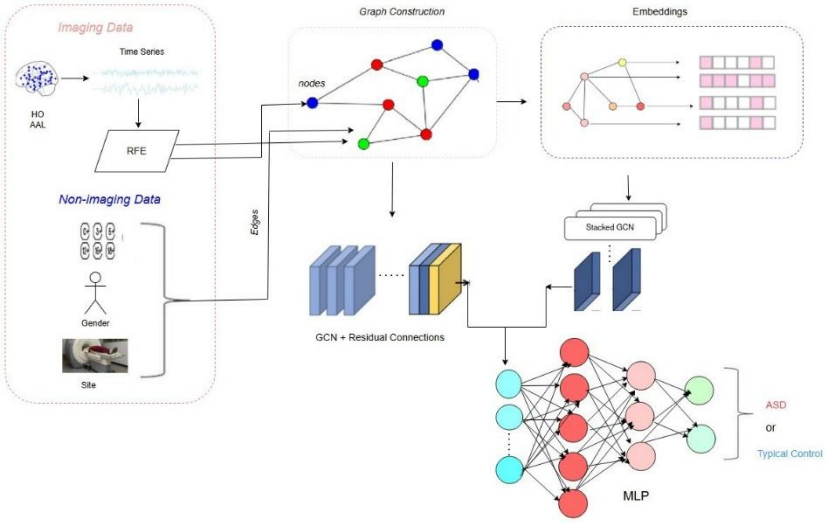

Fig. 1. Proposed Architecture.

3 Methodology

A schematization of the methodology to be used is presented in Fig 1. We adopt a graph-based approach to represent multimodal data pertaining to the diagnosis of Autism Spectrum Disorder (ASD) as a node-classification problem. Functional connectivity networks are constructed using fMRI time-series data, parcellated with the AAL and HO atlases [20] defining the graph nodes are brain regions and the edges are pairwise correlations of their time-series signals. Non-image data, such as age, gender, site and FIQ, are added for increasing the graph connectivity by capturing demographic similarities between the subjects. As shown in Fig 1, a Recursive Feature Elimination (RFE) mechanism is applied to select the most discriminative nodes and edges, optimizing the graph structure by iteratively removing the least important features [21]. The refined graph is then propagated to a Graph Convolutional Network (GCN) with residual connections to learn initial node embeddings, capturing local and global topological patterns in brain connectivity [22]. From the same graph, node

embeddings are extracted, and random walk embeddings are constructed to capture structural proximity between nodes using a random walk-based approach. These random walk embeddings are then propagated to a stacked GCN, which further refines the representations by modeling higher-order relationships. The output layers of the GCN with residual connections and the stacked GCN are combined, integrating both sets of learned features. Finally, the combined features are forwarded to a Multi-Layer Perceptron (MLP) for classification, outputting predictions as ASD or Typical Control.

3.1 Data Acquisition and Preprocessing

We conducted our experiments using the Autism Brain Imaging Data Exchange (ABIDE) dataset, which is openly accessible. multimodal repository [23]. ABIDE-I includes 1112 subjects from 17 global sites, providing resting-state fMRI (rs-fMRI) and phenotypic data [24]. We used a preprocessed version from the Preprocessed Connectomes Project, processed via the C-PAC pipeline. To ensure quality, we excluded subjects with missing time-series, incomplete brain coverage, severe motion, or artifacts, resulting in 871 subjects: 403 with ASD and 468 typical controls. For a balanced dataset, we selected 806 subjects, split equally into 403 ASD and 403 non-typical controls. The rs-fMRI data was parcellated using two atlases: the Harvard-Oxford (HO) atlas, yielding 111 ROIs [25], and the Automated Anatomical Labeling (AAL) atlas, yielding 116 ROIs. Average time-series for each ROI from both atlases were extracted to construct functional connectivity networks. Non-imaging data, including gender, were encoded (e.g., gender as binary, site as one-hot) to capture demographic variations**.**

3.2 Graph Nodes & Graph Edges Construction

In the constructed population graph G = (V, E, A, X), each node $v_i \in V$ corresponds to a distinct subject, and the associated node features are derived from resting-state functional MRI (fMRI) data. For each subject *i*, a functional connectivity (FC) matrix $F_i \in R^{R\times R}$ is computed by evaluating the Pearson correlation between the BOLD time series of each pair of brain regions (ROIs),

where *R* denotes the total number of ROIs defined by an anatomical atlas such as the Harvard-Oxford (HO) atlas and the Automated

Anatomical Labeling (AAL) atlas. Given that each FC matrix F_i is symmetric, we remove the redundant upper-triangular and diagonal elements, and flatten the strictly lower-triangular part to form the node feature vector:

$$\boldsymbol{x_i} = \boldsymbol{vec_{lower}}(\boldsymbol{F_i}) \in \boldsymbol{R^M} \quad (1)$$

Where $\boldsymbol{M} = \frac{\boldsymbol{R(R-1)}}{\boldsymbol{2}}$ (2)

Unlike conventional approaches that define inter-node similarity purely based on neuroimaging features. we construct graph edges by leveraging subject-level demographic attributes, including age, gender, and data acquisition site. Each subject *i* is associated with

a demographic vector $\mathbf{d}_i \in \mathrm{R}^D$, where D is the number of demographic attributes. To quantify the similarity between two subjects i and j, we compute the cosine similarity between their demographic vectors:

$$\boldsymbol{A_{ij}} = \boldsymbol{cos}\,(\boldsymbol{d_i}, \boldsymbol{d_j}) = \frac{\boldsymbol{d_i^T d_j}}{||\boldsymbol{d_i}||_2 \cdot ||\boldsymbol{d_j}||_2} \tag{3}$$

where $\mathrm{d}i, \mathrm{d}j \in \mathrm{R}^D$ denote the demographic vectors for subjects i and j, respectively. This similarity score is then used as the edge weight in the population graph, enabling the model to capture learned demographic relationships among subjects through data-driven edge construction. This formulation ensures that edge weights reflect normalized demographic affinities between subjects. To enhance the sparsity and scalability of the graph, a k-nearest neighbor (k-NN) sparsification scheme is applied such that each node retains edges only to its k most similar neighbors:

$$\boldsymbol{A_{ij}} = \begin{cases} \boldsymbol{cos}(\boldsymbol{d_i}, \boldsymbol{d_j}), & \boldsymbol{if\ j \in Top_k(i)} \\ \mathbf{0}, & \boldsymbol{otherwise} \end{cases} \tag{4}$$

This leads to a sparse affinity matrix $\mathrm{A} \in \mathrm{R}^{N \times N}$ and completes the definition of the population graph G, which encodes both imaging-based features and demographic similarity structure.

3.3 Random Walk-Based Graph Embedding: DeepWalk

To capture global structural information that encompasses both local neighborhood relationships and long-range dependencies that may reflect temporal or developmental similarities among subjects, we employ the DeepWalk algorithm a stochastic embedding method that uses truncated random walks and a skip-gram model to learn continuous vector representations of graph nodes.

Let $G = (V, E)$ be an undirected graph, where V is the set of nodes that represent subjects and $E \subseteq V \times V$ denotes edges representing demographic similarity. DeepWalk aims to learn an embedding function $\Phi: V \rightarrow \mathrm{R}^d$ that maps each node $v \in V$ to a low-dimensional vector $\mathrm{z}v \in \mathrm{R}^d$, capturing both local and global structural information.

1. Random Walk Generation: For each node v ∈ V, we simulate γ truncated random walks of fixed length L, resulting in walk sequences:

$$w_v^{(i)} = (v_0 = v, v_1, v_2, \ldots, v_{L-1}) \tag{5}$$

$$Where\ v_t + 1$$

Here, $\mathrm{N}(v_t)$ denotes the neighbors of node v_t, and U denotes the uniform distribution over those neighbors. The total corpus of walks is:

$$\mathcal{W} = \bigcup_{v \epsilon V} \bigcup_{i=1}^{\gamma} \mathcal{W}_v^{(i)} \tag{6}$$

Skip-Gram Embedding Learning*:* Each walk W ∈ W is treated analogously to a sentence in a language model. Using a Skip-gram architecture, we aim to maximize the likelihood of observing context nodes within a window size *w* around a central node. The optimization objective is:

$$\max_{\Phi} \sum_{w \epsilon W} \sum_{t=1}^{|W|} \sum_{-w \le j \le w, j \neq 0} log\, P(v_{t+j} | z_{v_t}) \tag{7}$$

where the conditional probability is modeled using softmax:

$$P(v_j | z_{v_i}) = \frac{exp\,(z_{v_j}{}^T z_{v_i})}{\sum_{v_k \epsilon v} exp(z_{v_k}{}^T z_{v_i})} \tag{8}$$

Due to computational costs, this softmax is approximated via negative sampling.The final output of the DeepWalk model is an embedding matrix:

$$Z = [\, z_1, z_2, \ldots\ldots, z_{|V|}]^T \in \mathbb{R}^{|V| x\, d} \tag{9}$$

where each row z_v represents the learned embedding for node $v \in V$.

3.4 Graph-Based Classification: Stacked GCN & DeepGCN

1) Deep Graph Convolutional Network (DeepGCN): To model complex topological structures and subject-level dependencies directly from graph-structured inputs, we adopt a Deep Graph Convolutional Network (DeepGCN) framework with residual connectivity. This architecture enables deeper propagation of node features without suffering from the issues of vanishing gradients and feature oversmoothing, which typically arise in standard GCNs.

We construct a population-level undirected graph G = (V, E), where each node $v_i \in$ V denotes a subject, and edges $e_{ij} \in$ E capture phenotypic similarity. A feature vector is associated with each node. $x_i \in R^C$, which yields a feature matrix $X \in R^{N \times C}$. The adjacency matrix $A \in R^{N \times N}$ encodes the connectivity, and D denotes its degree matrix. To enable localized spectral filtering, we adopt the Chebyshev polynomial approximation of graph convolutions. The l^{th} hidden layer output $y^{(l+1)}$ is computed as:

$$y^{(l+1)} = g(D^{-1/2} A D^{-1/2} y^{(l)} W^{(l)} \tag{10}$$

where $y^{(0)} = x$, $W^{(l)}$ is the trainable weight matrix, and G (·) is a non-linear activation function (e.g., ReLU). This formulation approximates spectral convolutions using K-order Chebyshev polynomials $T_k(\tilde{L})$ of the normalized Laplacian $\tilde{L}$, which avoids explicit eigen-decomposition.

However, stacking multiple GCN layers introduces gradient instability, where feature representations collapse to indistinguishable vectors across nodes. To alleviate this, we integrate residual connections, inspired by Reset, into the GCN architecture. The residual formulation converts the transformation $H(y^{(l)})$ into:

$$y^{(l+1)} = S(y^{(l)}, W^{(l)}) + y^{(l)} \quad (11)$$

Here, S (·) denotes graph convolution operation with learnable parameters $W^{(l)}$. Each layer's output is thus a summation of its transformation and the original input, promoting gradient flow and mitigating feature degradation.

All intermediate convolutional layers incorporate ReLU activations and dropout regularization to further stabilize training and reduce overfitting. Notably, residual units are excluded in the final layer, such that the output feature representation $\hat{X} \in R^{N \times P}$ is solely derived from the last graph convolution, before being passed to a Multi-Layer Perceptron (MLP) for classification.

This DeepGCN branch complements the Stacked GCN by directly leveraging the raw graph structure and node-level features in a deeper, more expressive architecture capable of modeling non-local and higher-order interactions across the population graph.

Stacked GCN: Traditional GCNs typically assume rich, high-dimensional input features for each node. However, in **our** case, the node attributes are sparse or unavailable. The StackedGCN addresses this issue by replacing raw features with unsupervised DeepWalk embeddings, enabling the model to learn over the structure of the graph rather than node-specific input.

The embeddings $Z \in R^{|V| \times d}$ obtained via the DeepWalk-based RWGE-DW framework are used as input node features for a stacked Graph Convolutional Network (GCN), which learns task-specific representations for subject-level classification. Let $G = (V, E)$ be the same demographic similarity graph as:

- $A \in R^{|V| \times |V|}$ denotes the binary or weighted adjacency matrix.
- $Z \in R^{|V| \times d}$ is the node feature matrix obtained from DeepWalk embeddings.
- $\hat{A} = A + I$ add self-loops.
- $\hat{D} \in R^{|V| \times |V|}$ is the diagonal degree matrix of $\hat{A}$

We stack L GCN layers to progressively refine node-level representations. The propagation rule for layer $\ell = 1, \ldots, L$ is:

$$\boldsymbol{H^{(l)} = \sigma(\hat{D}^{-1/2}\hat{A}\hat{D}^{-1/2}H^{(l-1)}W^{(l)}} \quad (12)$$

- $W^{(\ell)} \in R^{d\ell - 1 \times d\ell}$ is the trainable weight matrix for the ℓ-th layer,
- $\sigma(\cdot)$ is an activation function, e.g., ReLU.

The final layer output $H^{(L)} \in R^{|V| \times d'}$ provides the task-specific node embeddings. For node

classification.

Final Classification: We combine features from two different graph models—one based on brain connectivity and the other on subject similarities. By merging their outputs, we capture richer information about each subject, which is then used by an MLP to make the final ASD prediction.

Let $\hat{X}^{stacked} \in R^{N\times d1}$ denote the output node embeddings obtained from the Stacked GCN branch, and let $\hat{X}^{deep} \in R^{N\times d2}$

represent the embeddings from the DeepGCN branch. These embeddings are concatenated along the feature dimension to form a

unified node representation:

$$\hat{\boldsymbol{X}} = \text{Concat}\,(\hat{\boldsymbol{X}}^{\boldsymbol{stacked}}, \hat{\boldsymbol{X}}^{\boldsymbol{deep}}) \in \mathbb{R}^{\boldsymbol{Nx\,(d_1+d_2)}} \quad (13)$$

This fusion preserve both the local message-passing hierarchy captured by the Stacked GCN and the global residual-enhanced representations from the DeepGCN. The resulting joint embedding. The shared Multi-Layer Perceptron (MLP) then processes $\hat{X}$ for node-level classification:

$$z = \text{MLP}\,(\hat{X}) \in R^{N\times C} \quad (14)$$

where $z_i \in R^C$ denotes the probability vector of the class for the subject i and C is the number of output classes (e.g., ASD vs. TD). The MLP consists of fully connected layers with nonlinear activation (e.g., ReLU), optional dropout for regularization, and a final softmax layer to produce probabilistic predictions.

This fusion paradigm allows the model to exploit diverse relational information across modalities and population views, thereby enhancing classification performance and robustness to graph construction variability.

4 Experiments and Results

We evaluated our proposed dual-stream GNN model on the ABIDE-I dataset using a nested 10-fold cross-validation scheme to ensure robust and unbiased performance estimation. The primary loop partitions the data is divided into training and testing subsets, while the inner loop further divides the training data into training and validation subsets for hyperparameter tuning and early stopping. This nested strategy ensures that the test set remains completely unseen during model development, preventing feature peeking and improving generalization. In each fold, node features were constructed from raw imaging data using functional connectivity profiles, while edge features were derived from phenotypic metadata (age, gender, and site). Functional graphs were passed to the DeepGCN stream, and phenotypic graphs were processed via DeepWalk to obtain random walk-based node embeddings, which were then fed into a StackedGCN. The two graph streams were fused at the feature level and classified using

a shared MLP. During training, model optimization was carried out using the Adam algorithm, configured with a learning rate of 0.001 and L2 regularization to prevent overfitting. To enhance generalization, early stopping was employed based on validation loss, with a patience threshold of 30 epochs. The model was trained for up to 250 epochs per fold. At each epoch, we computed accuracy, AUC, precision, recall, and F1-score on both the training and validation sets. The best model for each fold was saved based on validation accuracy and used for final evaluation on the corresponding test set. We explicitly evaluate the role of edge pruning via edge dropout on model generalization. During training, edge weights in the connectivity graph are randomly dropped with a probability p = 0.2, in order to prevent overfitting and induce robustness against noisy or spurious correlations.

Algorithm 1 Pseudocode

Require: Functional signals X, demographic features D, number of GCN layers G_h, mapping functions S
Ensure: Predicted class label $\hat{Y}$ for each subject

1. Functional features X are processed to obtain X′
2. X′ and D are used to construct two graphs: G_{FC} (functional) and G_{RW} (phenotypic)
3. Node embeddings Z_{RW} are computed using random walk encoder from G_{RW}
4. Z_{RW} is passed through a stacked GCN to obtain $Zstacked$
5. $z_{deep}(0)$ = X' is used as input for DeepGCN
6. for gh=1gh = 1 to G_h do
7. if g_h= 1 then
8. $z_{deep}{}^{(gh)} = \mathcal{S}\,(z_{deep}{}^{(gh-1)}, W^{(gh)})$
9. else
10. $z_{deep}{}^{(gh)} = \mathcal{S}\,(z_{deep}{}^{(gh-1)}, W^{(gh)}) + z_{deep}{}^{(gh-1)}$
11. end if
12. end for
13. Concatenate representations: Z= $[z_{stacked} \,||\, z_{deep}{}^{(G_h)}$
14. Feed Z into an MLP: $\hat{Y} = MLP(Z)$

4.1 Performance Evaluation

To assess the effectiveness of the model, we utilize several evaluation metrics, including accuracy (ACC), precision, recall, F1 score, and the Receiver Operating Characteristic (ROC) curve. A true positive (TP) indicates a correctly identified positive instance, while a true negative (TN) represents an accurate classification of a negative case. Conversely, a false negative (FN) refers to a positive instance that has been incorrectly labeled as negative. These evaluation criteria are formally defined below:

$$\text{ACC} = \frac{\mathbf{TP+TN}}{\boldsymbol{TP+TN+FP+FN}} \tag{15}$$

$$\text{Precision} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FP}} \tag{16}$$

$$\text{Recall} = \frac{TP}{TP+FN} \tag{17}$$

$$\text{F1} = \frac{2x\ \text{Precision x Recall}}{\text{Precision+Recall}} \tag{18}$$

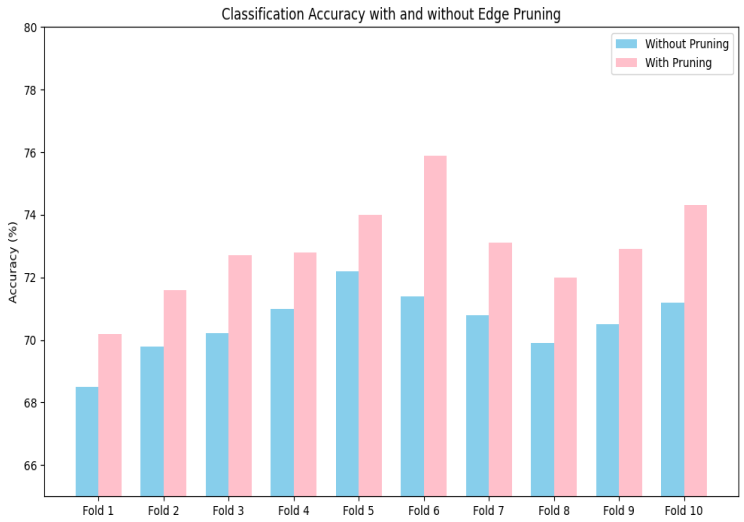

Fig. 2. Classification Accuracy with and without Edge Pruning.

4.2 Effect of Edge Pruning on Model Performance

To investigate the impact of edge pruning on graph-based learning, we incorporated an edge pruning mechanism into our graph- convolutional network (GCN) framework. Specifically, we trained the model under two configurations one with edge pruning applied during training and the other without it. The pruning mechanism estimates the importance of each edge based on the magnitude of the learned attention or the weights of the edges. At regular intervals, dynamically determined during training, edges that are less important than a computed threshold was removed. This threshold was adaptively set as the average value of all edge importance scores at that stage.

This selective pruning process is intended to remove noisy or weakly contributing edges, thereby simplifying the graph structure and reducing the potential for overfitting. It encourages the model to focus on the most relevant and informative connections. During training, we monitored key performance metrics such as classification accuracy, area under the ROC curve (AUC), and F1 score using 10-fold cross-validation setup. It was consistently observed that using edge pruning resulted in improved validation performance across multiple folds. One particularly valuable feature of our approach is that the interval at which pruning occurs is updated adaptively it increases when the validation loss decreases, allowing the model to stabilize, and decreases when validation performance degrades, promoting further exploration of the edge space. We also enforced a constraint that ensures that the number of remaining edges in the graph does not drop below a minimum threshold. This guarantees sufficient connectivity for effective message transmission between nodes.

Fig-2 presents a bar plot comparison of classification accuracies for selected folds, with and without edge pruning. For example, in Fold 3, the accuracy improved from 70.21% to 72.7%, while in Fold 6, it increased from 71.4% to 75.9%. On average, edge pruning yielded a 2.3% improvement across all folds. This trend indicates a more consistent and robust generalization performance when pruning is applied. The integration of adaptive edge pruning introduces a powerful form of structural regularization. It helps the model focus on semantically and functionally meaningful relationships in the graph, leading to more robust and generalizable learning, particularly in domains such as functional brain connectivity, where noisy correlations are prevalent.

4.3 Impact of Network Depth on Model Performance

In order to explore how architectural depth affects model performance, we performed a series of experiments by varying the number of graph convolutional layers within our framework. Specifically, we evaluated configurations with 2, 8, 16, and 32 layers in both the stacked GCN architecture with residual connections and the GCN framework. Each configuration was trained on two datasets, AAL and HO, with differing graph structures and feature complexities, enabling the evaluation of depth robustness across data distributions. Performance was evaluated via 10-fold cross-validation for statistical reliability.

The motivation behind this investigation was to understand the trade-offs between representational capacity and over smoothing, a common phenomenon in deep graph neural networks where increasing depth can lead to indistinguishable node representations and degraded performance. Incorporating residual connections improved gradient flow and feature retention across layers. Models with 8 and 16 layers outperformed both shallow (2-layer) and deep (32-layer) variants, offering a strong trade-off between expressiveness and stability. The 8-layer model consistently generalized well, while the 16-layer setup offered marginal gains with added complexity.

Table 1. Performance Metrics of GCN Models on AAL And Ho Datasets (Averaged For 8 And 16 Layers)

Dataset	Accuracy	Precision	F1 Score	AUC
AAL	81.2	78.80	77.38	84.85
HO	77.38	77.55	77.05	83.50

In parallel, we experimented with the GCN architecture, which integrates residual connections, dilated convolutions, and normalization techniques designed to support deeper learning. Consistent with the stacked GCN findings, the GCN models exhibited superior performance at intermediate depths (8 and 16 layers). The 2-layer version was underpowered, lacking sufficient depth to model higher-order dependencies, whereas the 32-layer variant experienced diminished returns, likely due to over-smoothing and optimization challenges despite architectural adaptations. Table 1 shows the Performance Metrics of GCN Models on AAL and HO Datasets.

Across both architectures, 8-layer and 16-layer configurations yielded the strongest performance. The 8-layer DeepGCN model achieved the highest average accuracy of 80.12%, while the 16-layer variant followed closely with 77.23%. These results highlight the benefit of moderate depth in preserving expressive power without introducing over-smoothing or training instability.

4.4 Overall Performance

To effectively assess the performance and stability of our model, DualStream-GCN, we conducted comprehensive experiments across two widely adopted brain atlases; AAL and HO. We contrasted our model with a variety of traditional and up-to-date models, including traditional GCN, DNN, EV-GCN. Baselines are non-graph-based model, shallow model, and new multimodal graph learning methods.For comparison fairness and reproducibility, we used nested 10-fold cross-validation, which splits the test data from the feature selection and training procedure, thus giving a more unbiased estimation of generalization performance.

The DualStream-GCN model incorporates both imaging and non-imaging data streams into one diagnosis pipeline. Imaging data, derived from AAL and HO fMRI time series, undergo feature refinement, subsequently forming subject-specific functional connectivity graphs. These graphs are then used to compute two streams of embeddings. In the first stream, the raw graph data comprising node and edge-level phenotypic information is directly passed through a Residual Graph Convolutional Network (Res- GCN) to preserve low-level topological patterns and promote gradient flow through deeper networks. In parallel, the second stream computes high-level graph embeddings, which are input into a Stacked GCN module that learns to capture both local connectivity and global population-level topology, while also modeling temporal dynamics implicitly encoded within node relationships. These dual representations are fused and subsequently fed into a Multi-Layer Perceptron (MLP) for final classification into Autism Spectrum Disorder (ASD) or Typical Control (TC) categories.

Our model attained a classification accuracy of 81.29% and an AUC of 0.85, outperforming all considered baselines on both datasets. The superior performance can be attributed to several key design principles within DualStream-GCN: (1) the residual connections in the Res-GCN branch, which mitigate vanishing gradients and enable the training of deeper architectures; (2) the stacked GCN embedding branch, which facilitates hierarchical feature abstraction and global context integration; and (3) the multi- modal fusion, which leverages both imaging and non-imaging features (age, gender, site), improving diagnostic robustness. These components work synergistically to enhance the expressive power of the model and capture subtle neurodevelopmental deviations indicative of ASD.

Furthermore, we performed statistical significance analysis by applying Student's t-test at a significance threshold of $\alpha = 0.05$. The performance improvements observed with DualStream-GCN were found to be statistically significant ($p < 0.05$) when compared to the closest baselines such as EV-GCN and MVS-GCN. Compared to EV-GCN (75.37% accuracy, 0.79 AUC), DualStream-GCN improved classification accuracy by 5.92% and AUC by 6 percentage points. In contrast to ASD-DiagNet, which utilizes single- modality data, our framework demonstrated that integrating non-imaging demographic features leads to enhanced classification fidelity. Notably, Hi-GCN, despite its hierarchical representation, achieved 4.17% lower accuracy, further validating the efficacy of our dual-branch architecture and

feature fusion strategy.

Table 2. Comparison of Performance Between Our Proposed Model and Various State-Of-The-Art Models Using 10-Fold Cross-Validation on The Aal And Ho Datasets.

Method	ACC	Precision	Recall	AUC	Parameter (M)
GCN	70.45	68.09	74.47	73.22	0.1
DNN	68.27	71.64	78.02	64.47	3.34
ASD-DiagNet	70.04	69.05	70.73	71.39	4.67
EV-GCN	75.37	75.22	84.45	79.12	0.13
Proposed	81.29	78.7	77.28	80.13	0.22

Additionally, Table 2 provides a comparative analysis of computational efficiency and training time across models. It highlights that while models like ASD-DiagNet and Hi-GCN require more parameters and longer training times, DualStream-GCN achieves superior performance with fewer parameters and faster training, making it ideal for resource-constrained clinical settings.

The runtime and model complexity analysis further emphasize the practicality of DualStream-GCN. Our model maintains a compact parameter space (~0.22M parameters) and efficient training time (~48 minutes per fold), while outperforming models like ASD-DiagNet, which require significantly longer training time (up to 16 hours) and higher parameter counts. The computational efficiency combined with strong classification performance positions DualStream-GCN as a compelling solution for real-world ASD diagnostic support systems.

5 Conclusions

This study introduced DualStream-GCN, a multimodal graph-based framework for ASD diagnosis that combines imaging and non-imaging features through two complementary embedding streams. The Residual GCN captures detailed local connectivity, while the Stacked GCN extracts higher-order structural and temporal patterns. These embeddings are fused and passed to an MLP for final classification into ASD or TC categories.

Evaluated on the ABIDE dataset using AAL and HO brain atlases, our model outperformed established baselines in accuracy and AUC, all while remaining computationally efficient. The dual-stream design proved effective in capturing both local and global neurofunctional representations critical for robust ASD classification.

While results are promising, challenges like over-smoothing in deeper layers and limited sample diversity remain. Future work will explore improved graph regularization, data augmentation, and interpretability techniques to strengthen model generalization and clinical insight. Lastly, interpretability remains an open concern. We plan to integrate explainable AI techniques, such as node-wise attribution and graph saliency analysis, to uncover critical brain regions and support biomarker discovery.

In conclusion, DualStream-GCN offers a scalable and interpretable approach for multimodal ASD detection, opening pathways for more robust and practical diagnostic support tools.

References

[1] Ma, C., Li, W., Ke, S. et al. Identification of autism spectrum disorder using multiple functional connectivity-based graph convolutional network. Med Biol Eng Comput 62, 2133–2144 (2024). https://doi.org/10.1007/s11517-024-03060-9

[2] Zwaigenbaum, L., Bauman, M. L., Stone, W. L., Yirmiya, N., Estes, A., Hansen, R. L., McPartland, J. C., Natowicz, M. R., Choueiri, R., Fein, C., Kasari, C., Pierce, K., Buie, T., Carter, A., Davis, P. A., Granpeesheh, D., & Mailloux, Z. (2015). Early identification of autism spectrum disorder: Recommendations for practice and research. Pediatrics, 136(Supplement 1), S10–S40. https://doi.org/10.1542/peds.2014-3667C

[3] Reddy, R. O., Rangaswamy, K., Dhanya, D., Devi, B. R., & C, S. K. R. (2022). Statistical analysis and deep learning associated modeling for early stage detection of carcinoma. International Journal on Recent and Innovation Trends in Computing and Communication, 10(2s), 116–126. https://ijritcc.org/index.php/ijritcc/article/view/5918

[4] Falkmer T, Anderson K, Falkmer M, Horlin C. Diagnostic procedures in autism spectrum disorders: a systematic literature review. Eur Child Adolesc Psychiatry. 2013 Jun;22(6):329-40. doi: 10.1007/s00787-013-0375-0.

[5] C, S.K.R., M, S. A 1-D CNN-FCM model for the classification of epileptic seizure disorders. Neural Comput & Applic 35, 17871–17881 (2023). https://doi.org/10.1007/s00521-023-08665-z

[6] Masi A, DeMayo MM, Glozier N, Guastella AJ. An Overview of Autism Spectrum Disorder, Heterogeneity and Treatment Options. Neurosci Bull. 2017 Apr;33(2):183-193. doi: 10.1007/s12264-017-0100-y.

[7] Loth, E., Spooren, W., Ham, L. et al. Identification and validation of biomarkers for autism spectrum disorders. Nat Rev Drug Discov 15, 70 (2016). https://doi.org/10.1038/nrd.2015.7

[8] Parisot, S. et al. (2017). Spectral Graph Convolutions for Population-Based Disease Prediction. In: Descoteaux, M., Maier-Hein, L., Franz, A., Jannin, P., Collins, D., Duchesne, S. (eds) Medical Image Computing and Computer Assisted Intervention − MICCAI 2017. MICCAI 2017. Lecture Notes in Computer Science (), vol 10435. Springer, Cham. https://doi.org/10.1007/978-3-319-66179-7_21

[9] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang and P. S. Yu, "A Comprehensive Survey on Graph Neural Networks," in IEEE Transactions on Neural Networks and Learning Systems, vol. 32, no. 1, pp. 4-24, Jan. 2021, doi: 10.1109/TNNLS.2020.2978386.

[10] Del Casale A, Ferracuti S, Alcibiade A, Simone S, Modesti MN, Pompili M. Neuroanatomical correlates of autism spectrum disorders: A meta-analysis of structural magnetic resonance imaging (MRI) studies. Psychiatry Res Neuroimaging. 2022 Sep; 325:111516. doi: 10.1016/j.pscychresns.2022.111516.

[11] Parisot S, Ktena SI, Ferrante E, Lee M, Guerrero R, Glocker B, Rueckert D. Disease prediction using graph convolutional networks: Application to Autism Spectrum Disorder and Alzheimer's disease. Med Image Anal. 2018 Aug; 48:117-130. doi: 10.1016/j.media.2018.06.001.

[12] Ktena SI, Parisot S, Ferrante E, Rajchl M, Lee M, Glocker B, Rueckert D. Metric learning with spectral graph convolutions on brain connectivity networks. Neuroimage. 2018 Apr 1; 169:431-442. doi: 10.1016/j.neuroimage.2017.12.052.

[13] Yousefian A, Shayegh F, Maleki Z. Detection of autism spectrum disorder using graph representation learning algorithms and deep neural network, based on fMRI signals. Front Syst Neurosci. 2023 Feb 2; 16:904770. doi: 10.3389/fnsys.2022.904770.

[14] Li, J., Liu, M., Li, Y., & Wang, Y. (2021). Identification of autism spectrum disorder with functional graph discriminative network. Frontiers in Neuroscience, 15, 729937. https://doi.org/10.3389/fnins.2021.729937

[15] Yang C, Wang P, Tan J, Liu Q, Li X. Autism spectrum disorder diagnosis using graph attention network based on spatial-constrained sparse functional brain networks. Comput Biol Med. 2021 Dec; 139:104963. doi: 10.1016/j.compbiomed.2021.104963

[16] Khodatars M, Shoeibi A, Sadeghi D, Ghaasemi N, Jafari M, Moridian P, Khadem A, Alizadehsani R, Zare A, Kong Y, Khosravi A, Nahavandi S, Hussain S, Acharya UR, Berk M. Deep learning for neuroimaging-based diagnosis and rehabilitation of Autism Spectrum Disorder: A review. Comput Biol Med. 2021 Dec; 139:104949. doi: 10.1016/j.compbiomed.2021.104949.
[17] Zhou W, Sun M, Xu X, Ruan Y, Sun C, Li W, Gao X. Multipattern graph convolutional network-based autism spectrum disorder identification. Cereb Cortex. 2024 Mar 1;34(3): bhae064. doi: 10.1093/cercor/bhae064.
[18] Bandara D, Riccardi K. Graph Node Classification to Predict Autism Risk in Genes. Genes (Basel). 2024 Apr 1;15(4):447. doi: 10.3390/genes15040447.
[19] Atlam, ES., Aljuhani, K.O., Gad, I. et al. Automated identification of autism spectrum disorder from facial images using explainable deep learning models. Sci Rep 15, 26682 (2025). https://doi.org/10.1038/s41598-025-11847-5
[20] Cui W, Du J, Sun M, Zhu S, Zhao S, Peng Z, Tan L, Li Y. Dynamic multi-site graph convolutional network for autism spectrum disorder identification. Comput Biol Med. 2023 May; 157:106749. doi: 10.1016/j.compbiomed.2023.106749.
[21] Shao, L., Fu, C. & Chen, X. A heterogeneous graph convolutional attention network method for classification of autism spectrum disorder. BMC Bioinformatics 24, 363 (2023). https://doi.org/10.1186/s12859-023-05495-7
[22] Nafisah, I., Jiang, Y., & Liao, B. (2025). Deep learning-based feature selection for detection of autism spectrum disorder. Frontiers in Artificial Intelligence, 8, 1594372. https://doi.org/10.3389/frai.2025.1594372
[23] Wang, C., Liu, Y., & Zhang, H. (2024). A novel approach for ASD recognition based on graph attention networks. Frontiers in Computational Neuroscience, 18, 1388083. https://doi.org/10.3389/fncom.2024.1388083
[24] Rubio-Martín, S., García-Ordás, M.T., Bayón-Gutiérrez, M. et al. Enhancing ASD detection accuracy: a combined approach of machine learning and deep learning models with natural language processing. Health Inf Sci Syst 12, 20 (2024). https://doi.org/10.1007/s13755-024-00281-y
[25] Gautam, S., Singh, R., & Sharma, S. (2023). Screening autism spectrum disorder in children using deep learning approach: Evaluating the classification model of YOLOv8 by comparing with other models. arXiv. https://doi.org/10.48550/arXiv.2306.14300