Knowledge Distillation on Cell Graphs for Node and Graph Level Classification

Presenters: Vasundhara Acharya and Hannah Powers
Group 1

Outline

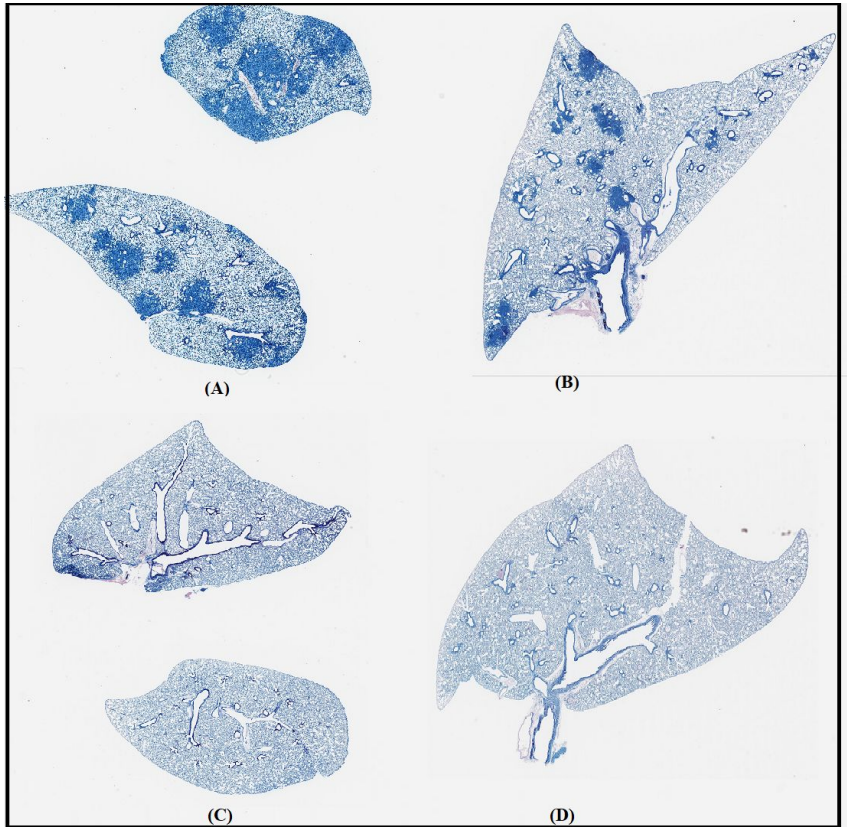
- Introduction
- Cell Graphs - A Solution
- Related works
- Gaps in Existing work
- Methodology
- Experimental Results and Ablation Studies
- Conclusion
- Future Scope

Introduction

- Pulmonary Tuberculosis (TB) is caused by Acid Fast Bacilli (AFB). It primarily affects the lungs and poses a severe health risk that cannot be ignored.
- The spatial patterns observed in the lung tissue microenvironment (TME) can provide valuable insights into the staging and progression of Tuberculosis.
- The average size of Whole slide images (WSI) that display these tissues are 42831*41159 at 40X magnification.
- Current deep learning methods typically rely on analyzing small image patches (for example ResNet takes 240*240 size images).

- Evaluating proteins in their full three-dimensional forms can be challenging due to their complex and intricate nature.
- Representing proteins as graphs simplifies this complexity.
- Graphs abstract the 3D spatial relationships into a simpler 2D form.
- Proteins are fundamental to virtually all biological processes.
- Enzymes are catalysts in biochemical reactions, essential for processes like metabolism, DNA replication, and signal transduction. Non-enzymes, on the other hand, have varied roles like structural support, transport, regulation, and defense.
- The classification into enzymes and non enzymes has various applications ranging from Drug Discovery and Therapeutics to Industrial Applications.

Sample Images in the Dataset



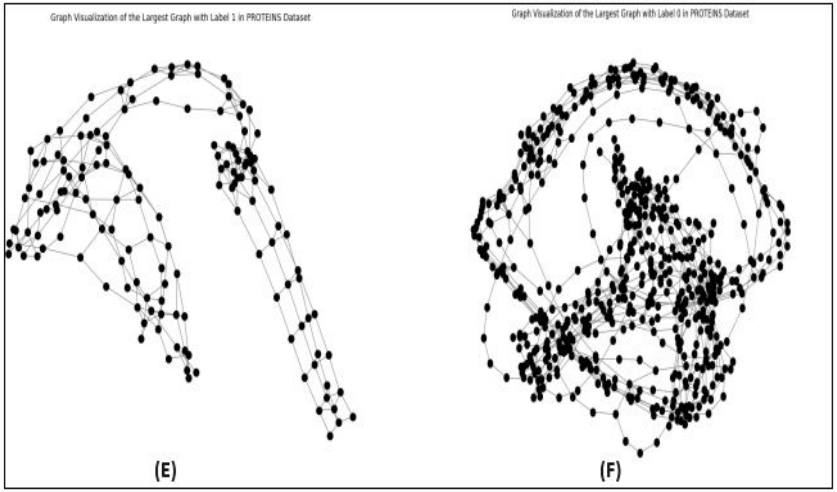


Figure : Sample images in the dataset. (A) and (B). TB infected. (C) and (D). Uninfected. (E). Enzyme. (F). Non Enzyme

Drawbacks of Patch-Wise Analysis of Whole Slide Images (WSI)

1. **Loss of Context**: Analyzing small patches can result in a loss of larger contextual information that is crucial for accurate diagnosis or understanding tissue architecture.

2. **Error Propagation**: Errors at the patch level can compound during aggregation, leading to inaccurate slide-level results.

3. **Resolution Challenges**: High-resolution WSIs may not fully capture important features like gland outlines when viewed at the patch level, leading to partial information and potential misclassification.

4. **Computational Constraints**: The large size of WSIs necessitates tiling and working with portions of the image, which can be computationally intensive and constrained by hardware capabilities.

Cell Graphs - A Solution

1. **Purpose:** To address limitations of patch-based methods and to avoid downsampling.
2. **How:** Convert the tissue structure in WSI into a graph representation:
   a. The nodes in the graph represent the different types of cells present in the tissue sample.
   b. The edges denote the interaction between these cells.
   c. A threshold ”d” indicates the edge threshold for intercellular communication.
   d. The correct setting of this threshold is crucial, as it ensures an accurate reflection of the biological interactions taking place.

Related Works: Cell Graphs

- A hierarchical cell-to-tissue-graph (HACT) model that, in comparison to existing models, closely resembled pathological diagnostic procedures and captured both cellular interactions and wide tissue morphology for detecting breast cancer was developed in [9].
- CGSignature, an AI-powered graph neural network approach utilizing spatial tissue microenvironment (TME) patterns from mIHC images to stage TME and predict patient survival in gastric cancer digitally, was proposed in [10]
- Bilgin et al. [11] proposed hierarchical cell graph for breast cancer tissue modeling.

Related Works: Knowledge Distillation in Graphs

We focus on knowledge distillation methods which distill logits for model compression

- MustaD compresses a k-layer graph convolution network (GCN) into a single-layer GCN by repeating the single layer k times and distilling the original model's logits and final node embeddings [14]
- Distill2vec used offline graph snapshots to train a teacher model whose logits were distilled to a student model tested on online graph snapshots [21]
- TinyGNN distills local structure knowledge and uses peer node information to learn the local structure [22]
- Jing et al [20] propose using two heterogeneous teacher models to distill their logits and node embeddings with a topological attribution map

Research Gaps

1. **Edge Threshold Decisions:**

Set without considering biological information of cells.
The threshold values chosen only care if the graph is sparse or dense

2. **Model Interpretability:**

Difficult to understand and interpret the results.
Lack of clarity on important features and their influence on predictions

3. **Lack of enough spatial information:**

The research works focus on simple spatial information such as the centroid of the cells.
They do not consider neighborhood overlap features which is crucial to understand the placement of the cells and how they interact with their neighboring cells.

4. **Graph Model Complexity:**

Models with millions of parameters.
Challenges in deploying for real-time medical inference.

Methodology

- Proposed a Jump Knowledge-Based Cell Graph Neural Network for enhanced learning.
- Adapted the number of layers in the teacher model to suit specific tasks, including node and graph classification.
- Trained a variety of student models to learn from the teacher model through basic knowledge distillation techniques.
- Employed a straightforward knowledge distillation approach by transferring logits from the teacher to the student models.
- Utilized integrated gradients for feature attribution in node classification tasks for interpretability.
- Comprehensive Evaluation Metrics: Assessed model performance using accuracy, F1 score, AUCPR scores, and distillation quality metrics.

DATASETS

Dataset for Node Classification (Lung Tissue Images):

- Source: Tuberculosis-infected lung tissue images from study [2].
- Data Size: 34 Whole Slide Images (WSIs) for training and validation, 10 WSIs for testing.
- Classification Goal: Classify nodes within the images into two categories - acid-fast bacilli (bacteria that causes TB) and nuclei.

Dataset for Graph Classification (Proteins Dataset):

- Source: Proteins dataset from study [3].
- Data Size: 1113 graphs representing proteins.
- Graph Composition: Nodes represent amino acids and edges denote interactions between them.
- Classification Goal: Classify each protein graph as either an enzyme or a non-enzyme.

Choice of Threshold -Cell Graphs

- Pathologists' insights are instrumental in refining these representations to better mirror actual cellular interactions [5].
- Mycobacterium tuberculosis bacteria within infected cells are very long, reaching a length of up to 150 micrometers after 72 hours of infection [6]
- The cells are not spherical and allow for a more significant detection as the macrophage extends pseudopods to sense its environment [7].
- Cells can extend part of their body (pseudopods) beyond their normal boundary (radius) to detect other cells that are farther away, allowing the detection range to exceed the standard limit of the cell’s radius[6].
- At a distance of about 6 Angstroms or less, amino acids can interact with each other through various types of non-covalent bonds like hydrogen bonds, ionic bonds, and van der Waals forces.

Adjacency Matrix -TB Graphs

$$A_{ij} \begin{cases} 1 & \text{if } Distance(u,v) < d \\ 0 & \text{otherwise.} \end{cases}$$

$$d(u,v) = \sqrt{(x_u - x_v)^2 + (y_u - y_v)^2}$$

TABLE III: Distance Thresholds

Node 'u'	Node 'v'	Distance 'd' in pixels
AFB	AFB	615
AFB	Nuclei	2049
Nuclei	AFB	2049
Nuclei	Nuclei	2049

Adjacency Matrix - Proteins Dataset

$$A_{ij} \begin{cases} 1 & \text{if } Distance(u,v) < d \\ 0 & \text{otherwise.} \end{cases}$$

$$d(u,v) = \sqrt{(x_u - x_v)^2 + (y_u - y_v)^2}$$

Where d =0.6 nanometers

Cell Graphs: TB Dataset

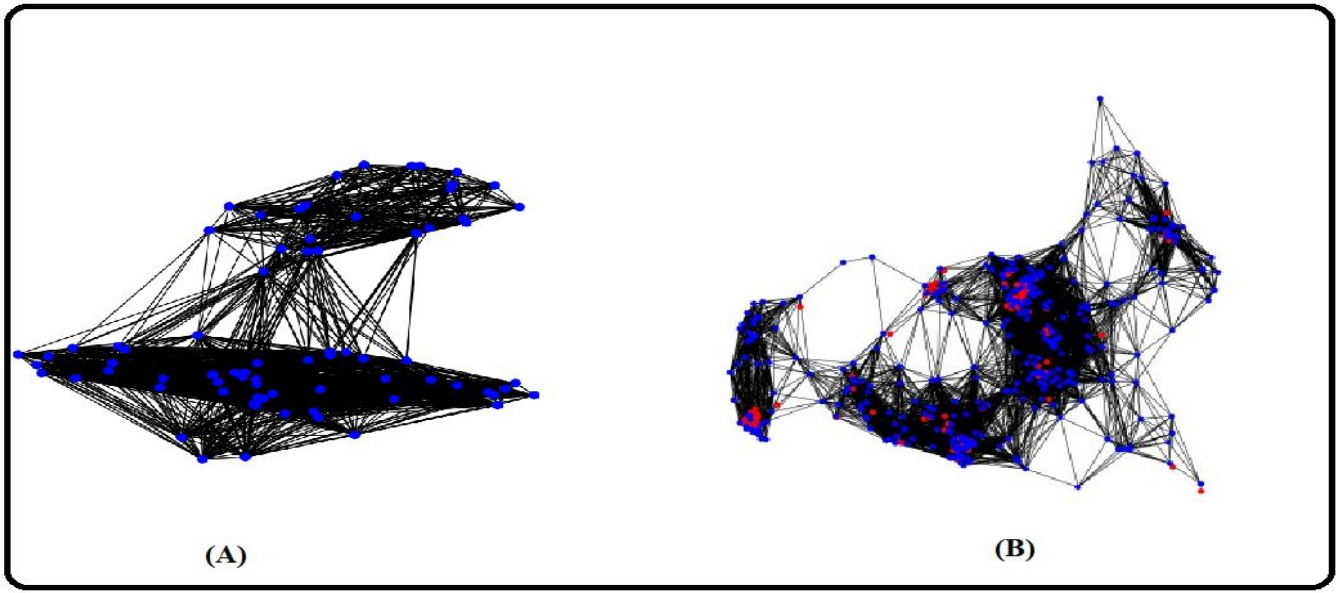
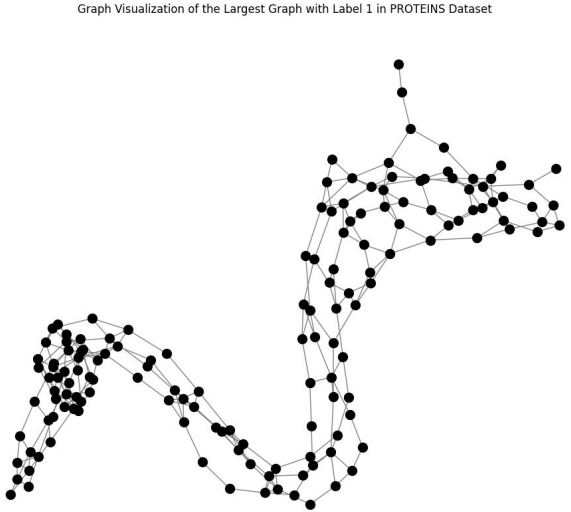


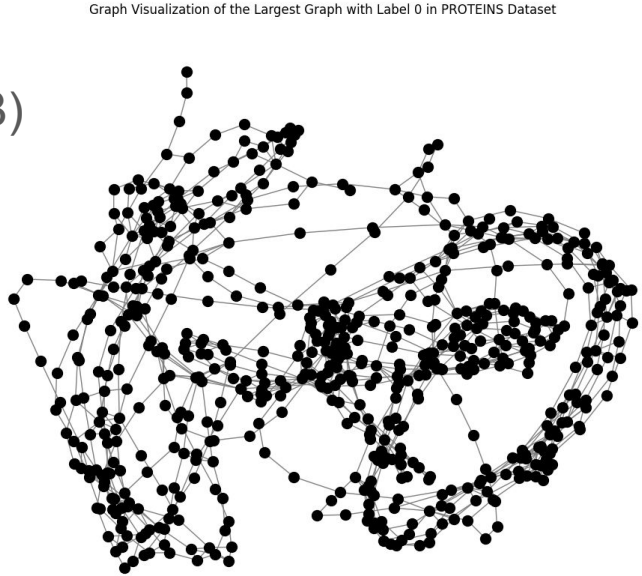
Figure 3: **Blue nodes indicate Nuclei. Red nodes denoted AFB. Black lines (edges) denote the interactions between the cells. (A). A cell graph of an uninfected sample. (B). A cell graph of an infected sample**

Cell Graphs: Proteins Dataset

(A)



(B)



(A). Graph with Label 1 (Enzyme). (B). Graph with Label 0 (Not Enzyme)

Our Approach: Knowledge Distillation (KD)

- The corresponding labels Y = {y1, y2, . . . , yn} signify the node-level labels for the graph data.
- The function ϕ is realized through the GNNs, transforming input graph data X, consisting of node features and edge indices, into non normalized outputs or logits.
- The non-normalized probability outputs of teachers and students are presented as zt = φ (X; Wt) and zs = φ (X; Ws). The distillation process employs a temperature scaling mechanism applied directly to these logits, formalized as :

$$p^{\tau} = \frac{\exp\left(\frac{z_i}{\tau}\right)}{\sum_j \exp\left(\frac{z_j}{\tau}\right)}, \qquad (6)$$

Our Approach: Knowledge Distillation

- The overall loss function for knowledge distillation can be succinctly expressed as follows

$$L_{KD} = \alpha L_{CE}(p_s, y) + \alpha \tau^2 KL(p_s^\tau, p_t^\tau) \qquad (7)$$

Here, L_CE (p_s,y) represents the cross-entropy loss. The second component, ατ^2 KL(p_s^τ,p_t^τ), is the knowledge distillation term. p_s^τ and p_t^τ denote the softened outputs of the student and teacher models, respectively, after applying the temperature scaling with parameter τ .

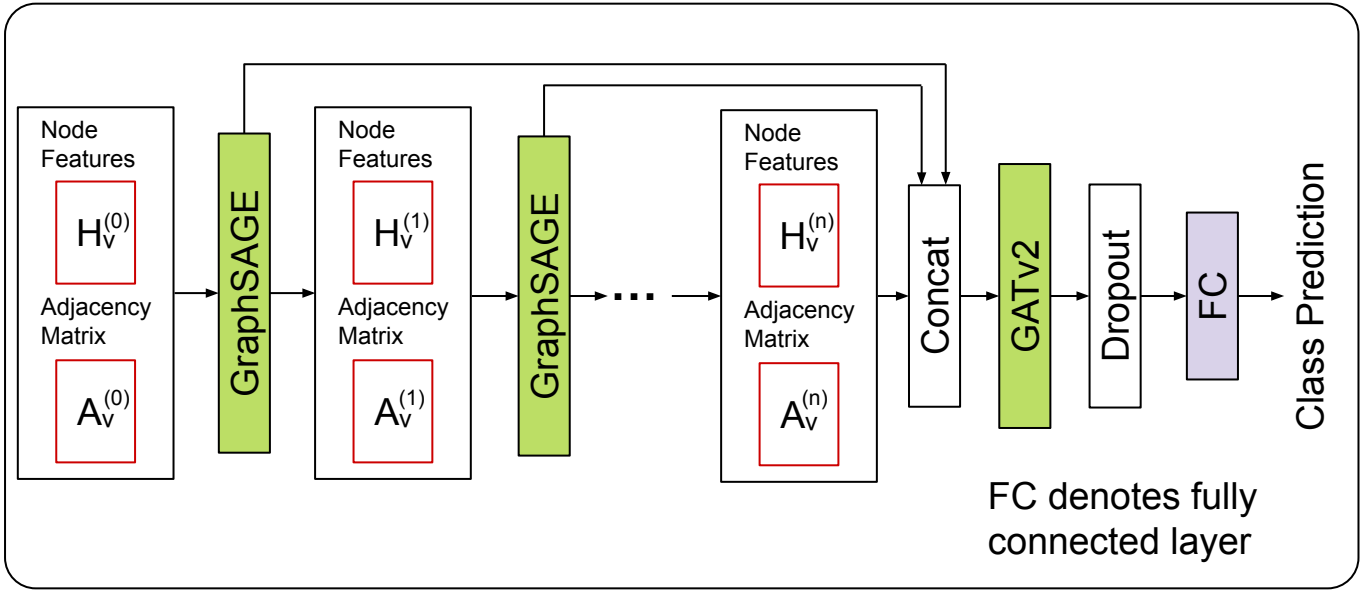
KL stands for the Kullback-Leibler divergence, a measure of how one probability distribution diverges from a second, reference probability distribution. α is a hyperparameter that controls the balance between the traditional cross entropy loss and the knowledge distillation loss.

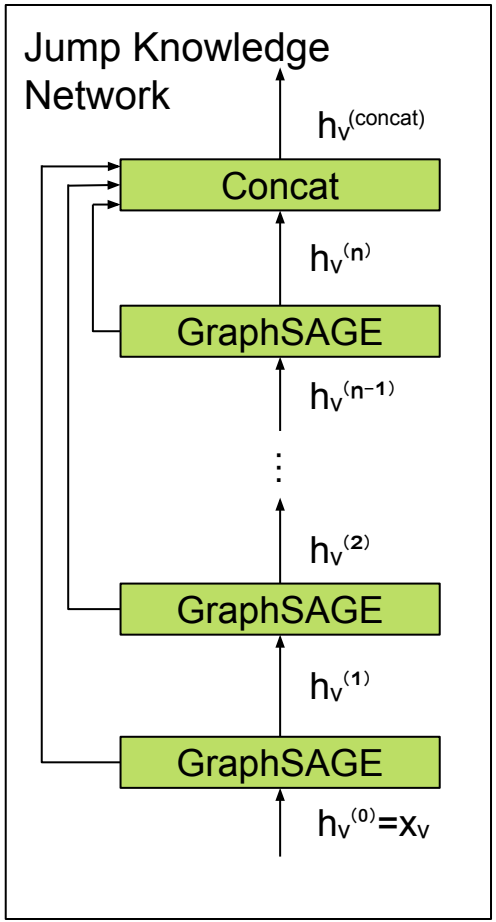
Research on Choice of Temperature and Alpha

- **Temperature Range for Softmax Function**: Hinton et al.[1] explored a range of temperatures from 1 to 20 for the softmax function in knowledge distillation. **They found that lower temperatures generally yield better results for smaller student models.**
- **Effect of Higher Temperatures**: Higher temperatures create a richer and more detailed distribution of soft labels. However, very small student models may struggle to benefit from this due to their limited capacity to capture subtle details in these soft labels.
- **Selection of Temperature Value**: In this study, a temperature value of 5 was chosen, considering the smaller size of the student models used and the findings of Hinton.
- **Weighted Average Technique for Loss Balanc**e: Hinton employed a weighted average method to balance distillation loss and student loss, using the relationship $\beta = 1 - \alpha$. Although they experimented with different values, the best results were usually achieved when α was significantly less than β.

- **Setting of α and β in the Current Study**: In contrast to other approaches, this study sets α to 0.5 to give equal importance to both distillation loss and the student model’s loss, different from some methods where α is set to 1 or no restrictions are placed on these parameters [4].
- The paper in [20] employed an adaptive temperature hyper-parameter, assigning a unique value to each node based on how confident the teacher model is about its prediction.
- They also found that including the teacher's entire prediction distribution in the temperature calculation is more effective than relying solely on confidence.

CG-JKNN Architecture





Oversmoothing handled using DropEdge technique
For graph classification: After the GATv2 layer, we would perform a global max pooling operation

Proposed Architecture

We propose a novel architecture for our teacher model: Cell Graph - Jumping Knowledge Neural Network (CG-JKNN)

- Uses GraphSAGE[12] layer to learn the nodes' hidden representation. GraphSAGE works by aggregating information from neighboring node representations

$$h_{N(v)}^{(l)} = \text{MEAN}\left(\left\{h_u^{(l-1)}, \forall u \in N(v)\right\}\right) \quad (1)$$

- The aggregated neighbor information is concatenated to the node information then multiplied with a weight matrix followed by the application of an activation function

$$h_v^{(l)} = \sigma\left(W \cdot \left[h_v^{(l-1)}, h_{N(v)}^{(l)}\right]\right) \quad (2)$$

- *W* is a learnable weight matrix

Proposed Architecture

- Introduce a “jumping knowledge knowledge representation”[13] capability into the model, done by concatenating all hidden representations of a node

$$h_v^{(Concatenated)} = \text{Concatenate}\left[h_v^{(1)}, \ldots, h_v^{(l)}\right] \quad (3)$$

- This new representation is then passed to a GATv2[15] layer which uses an attention mechanism:

$$\alpha_{vu} = \text{softmax}_u\left(\text{LeakyReLU}\left(\mathbf{a}^T\left[\mathbf{W}h_v^{(\text{Concatenated})} \| \mathbf{W}h_u^{(\text{Concatenated})}\right]\right)\right) \quad (4)$$

- The new node representation is then found with

$$h_v^{(GAT)} = \sigma\left(\sum_{u \subset \mathcal{N}(v)} \alpha_{vu}\mathbf{W}h_u^{(\text{Concatenated})}\right) \quad (5)$$

Features Extracted

Table 1: Local Graph Features: Employed for Node Classification

Feature	Description	Feature	Description
Eccentricity	The maximum graph distance between a vertex v and any other vertex u in a connected graph G .	Closeness_of_node	Denotes node's proximity to all other nodes in the network.
Average_clustering	Mean of local clustering of the graph.	Hub_promoted	Ratio of common neighbors of nodes a and b to the minimum of their node degrees .
Node_clustering	Degree to which nodes in a graph tend to cluster together.	Hub_depressed	Ratio of common neighbors of nodes a and b to the maximum of their node degrees .
Sorenson	Ratio of nodes u and v's common neighbors to their average node degrees.	Global_overlap	The number of all possible paths between two particular nodes.
Salton	Angle between columns of the adjacency matrix corresponding to the specified vertices	Mean_all_neighbors	The mean of the distance between a vertex v and all its neighbors in the graph G.
Kurtosis_all_neighbors	Kurtosis of edge lengths between a node v and all its neighbors.		

Table 2: Global Graph Features

Feature	Description	Feature	Description
Number_of_nodes	The number of nodes in the graph (represents the cells) .	Eigen_one_L	Number of eigenvalues of Laplacian matrix that have a value of one .
Number_of_edges	The number of edges in the graphs (interaction between the cells) .	Eigen_two_L	Number of eigenvalues of Laplacian matrix that have a value of two .
Radius	Minimum eccentricity	Lower_slope_A	Line segment's slope that corresponds to adjacency matrix's eigenvalues between 0 and 1.
Center	The group of nodes having an eccentricity equal to the radius.	Upper_slope_A	Line segment's slope that corresponds to adjacency matrix's eigenvalues between 1 and 2
Trace_A	Sum of the diagonal elements of the adjacency matrix from upper left to lower right.	Energy_A	Sum of absolute value of adjacency matrix's eigenvalues .
Upper_Slope_L	Line segment's slope that corresponds to Laplacian matrix's eigenvalues between 1 and 2.	Eigen_zero_A	Number of eigenvalues of adjacency matrix that have a value of zero.
Connected_ratio	Number of nodes in the graph's largest?connected component divided by the overall number of nodes.	Eigen_one_A	Number of eigenvalues of adjacency matrix that have a value of one.
Trace_L	Sum of its diagonal entries and the sum of its eigenvalues of Laplacian matrix .	Eigen_two_A	Number of eigenvalues of adjacency matrix that have a value of two.
Energy_L	Absolute value sum of Laplacian matrix's eigenvalues .	Node_degree_0	Number of nodes with degree zero .
Diameter	Maximum eccentricity	Node_degree_one	Number of nodes with degree one .
Lower_slope_L	Line segment's slope that corresponds to Laplacian matrix's eigenvalues between 0 and 1.		

Stages of Morphological Features Extraction

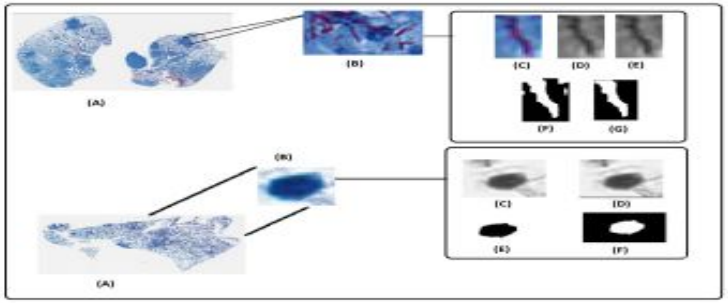


Figure 2: **Upper Row: (A). Original Image. (B). Region of Granuloma with AFB. (C). Single AFB at the location (pixel) (46200,12954)in the original image. (D). Grayscale image. (E). Enhanced image. (F). Binary image before post-processing. (G). AFB.**
Lower Row: (A). Original Image. (B). Single Nuclei at the location (pixel) (25424,16909) in the original image. (C). Grayscale image. (D). Enhanced image. (E). Image after morphological operations. (F). Nuclei.

Table 3: Shape and Texture Features

Features	Description
X	X coordinate of the cell center.
Y	Y coordinate of the cell center.
Contrast	Measures the local variations in the gray-level co-occurrence matrix.
Energy	Computes the sum of squared elements in the GLCM.
Correlation	Calculates the combined likelihood that the provided pixel pairs will occur.
Homogeneity	The degree to which the distribution of elements in the GLCM is close to the GLCM diagonal.
ASM Value	Measure of homogeneity of an image .
Dissimilarity	The distance between two objects (pixels) in the region of interest.
Variance	The gray level distribution's dispersion (with respect to the mean).
Mean Image	Ratio of sum of pixel values to the total number of pixel values .
Standard Deviation	Measure of image gray level intensity dispersion .
Area	Measures the actual number of pixels in the region.
Major Axis	Length (in pixels) of the ellipse's major axis that shares the same normalized second central moments as the region .
Minor_axis	Length (in pixels) of the ellipse's minor axis that shares the same normalized second central moments as the region .
Eccen	The eccentricity is determined by dividing the ellipse's major axis length by the distance between its foci.
Perimeter	Computes the distance around the region's border.
Diameter_x	Represents the mean of major axis and minor axis length.
Circularity	Computes the roundness of the object.
Mean_convex_hull	Mean of the group of pixels contained in the smallest convex polygon that encircles each white input pixel.
SD_convex_hull	Standard Deviation of the group of pixels contained in the smallest convex polygon that encircles each white input pixel.

Evaluation Metrics

- $$\text{Accuracy} = (TP+TN)/(TP+TN+FP+FN) \quad (8)$$

- $$\text{AUPRC} = \int_0^1 p(r)dr \quad (9)$$

  where $p(r)$ is the precision at recall r

- $$\text{F1-Score} = \frac{2}{\frac{1}{\text{Recall}} + \frac{1}{\text{Precision}}} \quad (10)$$

We measure the distillation quality with

- $$DS = \alpha \cdot \left(\frac{\text{student}_s}{\text{teacher}_s}\right) + (1-\alpha) \cdot \left(1 - \frac{\text{student}_{\text{f1}}}{\text{teacher}_{\text{f1}}}\right) \quad (11)$$

Experimental Set Up and Results

- We implemented the models using the PyTorch framework and ran them on one NVIDIA A100 GPU.
- Ensuring a systematic and fair comparison requires the optimization of hyperparameters for every model and test problem individually [23].
- Evaluated the model's performance with and without knowledge distillation across a comprehensive set of graph and morphological features.
- Demonstrated improvement in model performance post knowledge distillation using our proposed distillation score.
- Investigated the relationship between the number of model parameters and performance using the Pearson correlation coefficient.
- Discussed the biological significance of features deemed important by the model.

Feature Attribution: Model Interpretability

- Implemented the integrated gradients method to calculate feature importance, providing insights into which features significantly influence model predictions.
- Applied this technique separately for two distinct classes - AFB and nuclei - to understand class-specific contributions of features.
- This approach enhances the interpretability of our machine learning model, offering a clear visualization of how each feature affects the classification of AFB and nuclei.
- While trying to determine node feature attributions for graph-level classification, we encountered a limitation in PyTorch's current capabilities, which primarily support edge masks.
- Recognizing that edge importance may not align with domain experts' needs for interpretability, we have opted not to include this in our analysis.

Feature Attribution: Without Knowledge Distillation

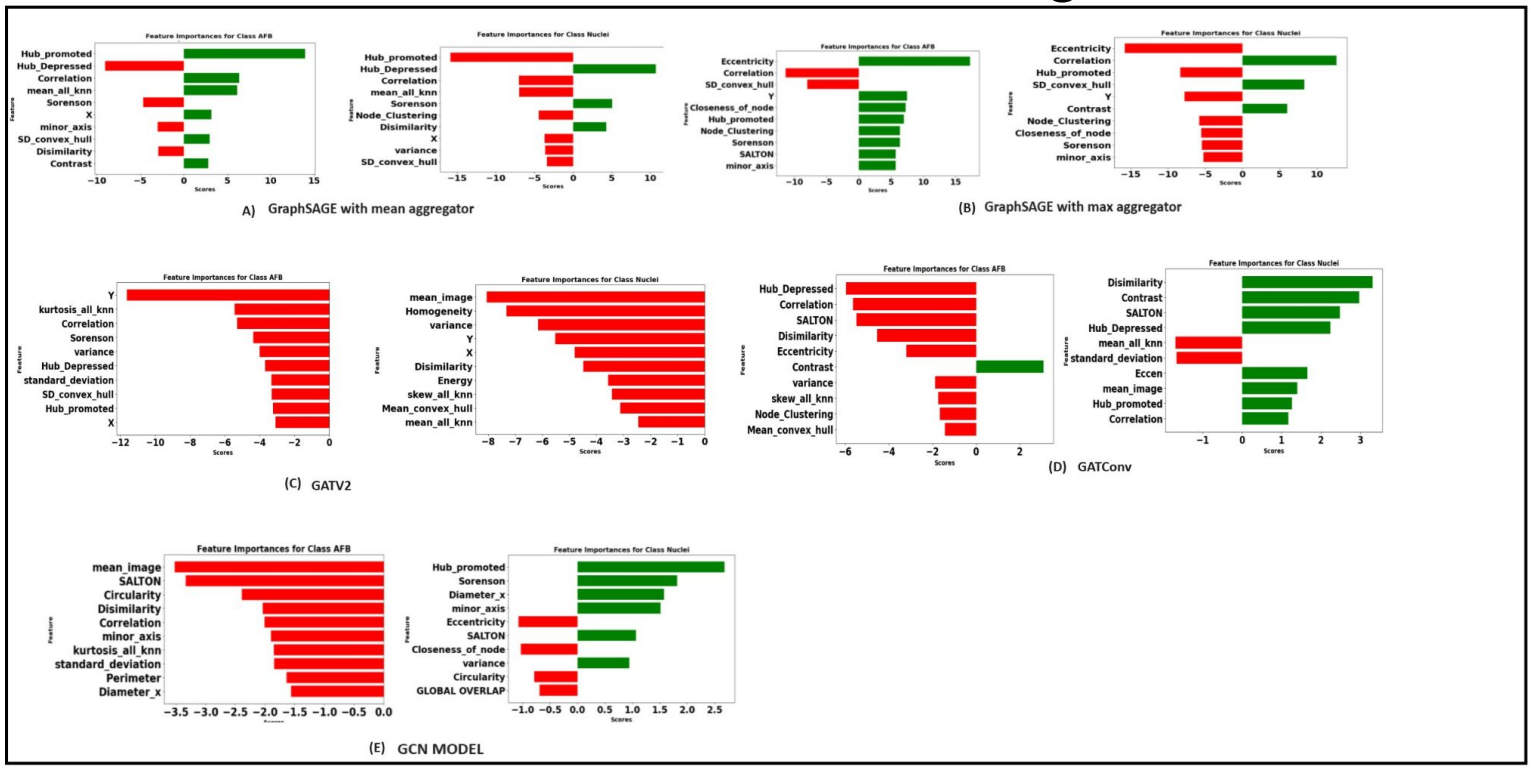


Figure 6: **Without Knowledge Distillation: Feature attribution scores of GNN models that performed best across three trials for class AFB and class Nuclei. (A). GraphSAGE with mean aggregator. (B). GraphSAGE with max aggregator. (C). GATV2. (D). GatConv. (E). GCN**

Feature Attribution: With Knowledge Distillation

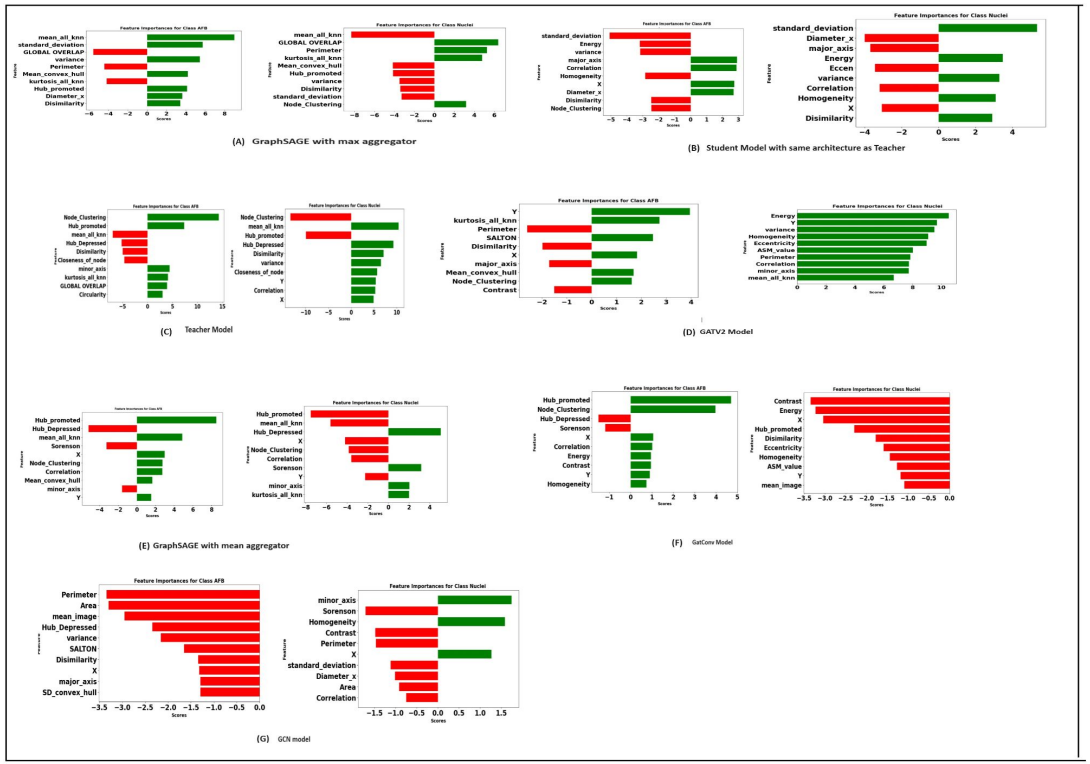


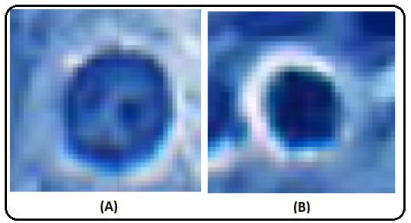
Figure 7: **With Knowledge Distillation: Feature attribution scores of GNN models that performed best across three trials for class AFB and class Nuclei. (A). GraphSAGE with max aggregator. (B). Student Model having same architecture as the teacher model. (C). Teacher Model (D). GATv2 (E). GATConv (F) GraphSAGE with mean aggregator. (F). GCN**

Interesting Observations

- Notably, a direct comparison between the teacher model (C) and the GATv2 model (D) reveals a discernible discrepancy in feature prioritization.
- For Class AFB, upon examining the feature attribution scores, we find that ’Node Clustering’ and ’Dissimilarity’ are the only features commonly identified as significant by both the GATv2 and the teacher model.
- This suggests a narrow overlap in feature prioritization for this class.
- In contrast, for Class Nuclei, there is a slightly broader consensus with features such as ’Variance’, ’Y coordinate of center’, and ’Mean all knn’ being consistently recognized as important by both models.
- The limited commonality in feature importance underscores the distinct learning patterns that GATv2 adopts, despite its overall strong performance in knowledge distillation.

Biological Relevance of Features

- We also see a higher node clustering coefficient for the node denoting AFB. It implies that the neighboring nodes of the node representing AFB are more likely to be connected, forming local clusters or communities. This can indicate a higher level of interconnectivity and cohesive structure around this node.
- The higher hub-promoted index suggests that the node denoting AFB strongly influences the overall network structure and information flow.
- We believe it resonates with the biological context as the host's inflammatory responses and immune system are triggered by the presence of the bacteria.
- Instances belonging to class AFB tend to exhibit higher contrast.
- This is mainly due to the staining procedure that involves using a red dye for AFB and a blue color for the other tissue.
- Higher variance in nuclei is due to the varying nuclear chromatin pattern.



Results Without Knowledge Distillation: Node Classification

Table 4: Combined Features: Node Classification Performance and Hyperparameters of Graph models without Knowledge Distillation. Model in bold is the best performing.

Model	Train_Acc	Val_Acc	Test_Acc	Train_F1	Val_F1	Test_F1	Hyperparameters
GraphSAGE_mean	95.136 ±0.07	90.183 ±0.069	86.656 ±0.209	0.9605 ±0.001	0.9532 ±0.003	0.8972 ±0.004	hidden_dim=78,lr=0.0054, dropout=0.1975, num_layers=2 Aggr=mean
GraphSAGE_max	**93.04** ±0.65	**88.57** ±0.655	**84.593**±0.499	**0.9261** ±0.015	**0.9111** ±0.025	**0.9092**±0.012	hidden_dim=80,lr=0.0054, dropout=0.1975, num_layers=2 Aggr=max
GATV2	93.30 ±0.28	89.68 ±0.88	83.78±1.013	0.9448 ±0.024	0.9359 ±0.014	0.8975±0.025	lr=0.012,heads=8, dropout=0.1, num_layers=2 weight_decay=5e-4
GatConv	90.38 ±1.28	87.38 ±1.34	81.73±1.136	0.9524±0.012	0.948 ±0.013	0.8818±0.031	lr=0.073,heads=8, dropout=0.111, weight_decay=0.0005, num_layers=2
GCN	51.36 ±0.071	50.94 ±0.721	61.86±0.041	0.9305±0.0268	0.9131 ±0.0729	0.9022±0.0016	lr=2.6637587897015506e-05, dropout=0.3514829925423625, , num_layers=2 , hidden_channels=42

Is all this Connectivity Required? Can't we just use positions of these cells or just morphology features?

TABLE 6: Morphology Features: Performance and Hyperparameters of ML models

Model	Train_Acc	Val_Acc	Test_Acc	Train_F1	Val_F1	Test_F1	Hyperparameters
XGBoost	**88.88**	**86.14**	**86.8**	**0.853**	**0.8174**	**0.829**	gamma=0.1,learning_rate=0.2, max_depth=5, estimators=100, reg_alpha=0.2, reg_lambda=0.3
Random Forest	85.29	82.99	83.58	0.8471	0.7786	0.7901	min_samples_leaf=1, min_samples_split=6, estimators=400
LightGBM	87.15	85.73	86.29	0.831	0.812	0.822	lr=0.1, max_depth: -1, min_child_samples: 20, num_leaves=31
Extra Trees	86.66	82.51	83.36	0.817	0.758	0.773	criterion=gini, min_samples_leaf=10, min_samples_split=5, estimators=100

***The poor performance of traditional ML models using only morphology features and coordinates compared to models that incorporated graph features suggests that the spatial relationships and connectivity between cells—information typically captured by graph-based features—are important for the node level classification.**

Results With Knowledge Distillation: Node Classification

Table 6: Combined Features: Node Classification Performance and Hyperparameters of Graph models with Knowledge Distillation

Model	Train_Acc	Val_Acc	Test_Acc	Train_F1	Val_F1	Test_F1	Hyperparameters
Teacher_Model (CG-JKNN)	96.82 ±1.48	91.42 ±0.88	86.74 ±0.99	0.9780 ±0.00978	0.9610 ±0.00627	0.9668 ±0.00445	lr=0.001, dropout=0.1, graph_dropout_prob=0.1 , num_sage_layers=25, num_gat_layers=1 , hidden_channels=33
GraphSAGE_mean	92.94 ±0.079	89.94 ±0.30	86.91 ±0.25	0.9541 ±0.00106	0.9524 ±0.001	0.9330 ±0.00113	hidden_dim=78,lr=0.0054, dropout=0.1975, num_layers=2 Aggr=mean
GraphSAGE_max	**92.14** ±0.34	**90.24** ±0.67	**83.31**±1.44	**0.9469** ±0.012	**0.9462** ±0.0018	**0.9361**±0.0123	hidden_dim=80,lr=0.0054, dropout=0.1975, num_layers=2 Aggr=max
GATV2	**91.76** ±0.58	**89.33** ±0.62	**82.20**±3.50	**0.95843** ±0.00133	**0.9689**±0.0025	**0.9517**±0.0022	lr=0.012,heads=8, dropout=0.1, num_layers=2 weight_decay=5e-4
GatConv	84.82 ±1.93	70.91 ±9.11	69.44±4.62	0.9554±0.0038	0.9659 ±0.00	0.9278±0.0064	lr=0.073,heads=8, dropout=0.111, weight_decay=0.0005, num_layers=2
GCN	50.43 ±0.39	49.02 ±1.11	59.41±1.3	0.9019±0.0723	0.9538±0.0085	0.8708±0.0618	lr=2.6637587897015506e-05, dropout=0.351482992542, , num_layers=2 , hidden_channels=42
CG-JKNN (smaller model)	**92.38** ±0.17	**90.46** ±0.067	**84.10**±2.36	**0.956**±0.0029	**0.9608** ±0.0041	**0.949**±0.0052	lr=0.001, dropout=0.1, graph_dropout_prob=0.1 , num_sage_layers=2, num_gat_layers=1 , hidden_channels=33

Table 8: The Impact of Knowledge Distillation on Model Performance: Node Classification

Model	Best Test F1 without knowledge distillation	Best Test F1 with knowledge distillation	Improvement (%)	Best Test AUCPR without knowledge distillation	Best Test AUCPR with knowledge distillation	Improvement (%)
GraphSAGE with max aggregator	0.9112	0.93413	2.45	0.9646	0.9882	2.44
GraphSAGE with mean aggregator	0.9212	0.9484	2.86	0.9753	0.9897	1.47
GATV2	**0.9225**	**0.9539**	**3.29**	**0.9709**	**0.9864**	**1.59**
GATConv	0.9128	0.9342	2.29	0.9484	0.9655	1.80
GCN	0.9438	0.9326	-1.20	0.8592	0.8801	2.43

Observations

- **Score Improvement**: With knowledge distillation, all models except GCN show an improvement in the Best Test F1 score, indicating better accuracy in classifying nodes correctly.
- **AUCPR Enhancement:** Similarly, knowledge distillation leads to an improvement in the Best Test AUCPR for all models, suggesting that the models became more precise and reliable in ranking the positive instances higher than negative ones.
- **Model Comparison**: GraphSAGE with mean aggregator and GATv2 exhibit the most significant improvements in F1 score, while GraphSAGE with max aggregator and GCN show the largest gains in AUCPR, highlighting the benefits and suitability of knowledge distillation for specific model architectures.

Table 10: Comparison of Model Complexity and Distillation Quality using F1 and AUCPR Scores : Node Classification.Model shown in bold is best performing

Model	Number_of_Parameters	Test AUCPR	Test F1 score	Distillation Quality with F1 score	Distillation Quality with AUCPR score
GraphSAGE with max aggregator	5682	0.9897	0.9484	0.037476651	0.026544671
GraphSAGE with mean aggregator	5682	0.9882	0.93413	0.044832321	0.028059976
GATV2	**1784**	**0.9864**	**0.95401**	**0.016513168**	0.01180661
GATConv	918	0.9655	0.93518	0.022204444	0.028904944
GCN	1514	0.8801	0.9326	0.026297489	0.117939433
Smaller version of CG-JKNN	6901	0.9883	0.9505	0.042045649	0.033610428

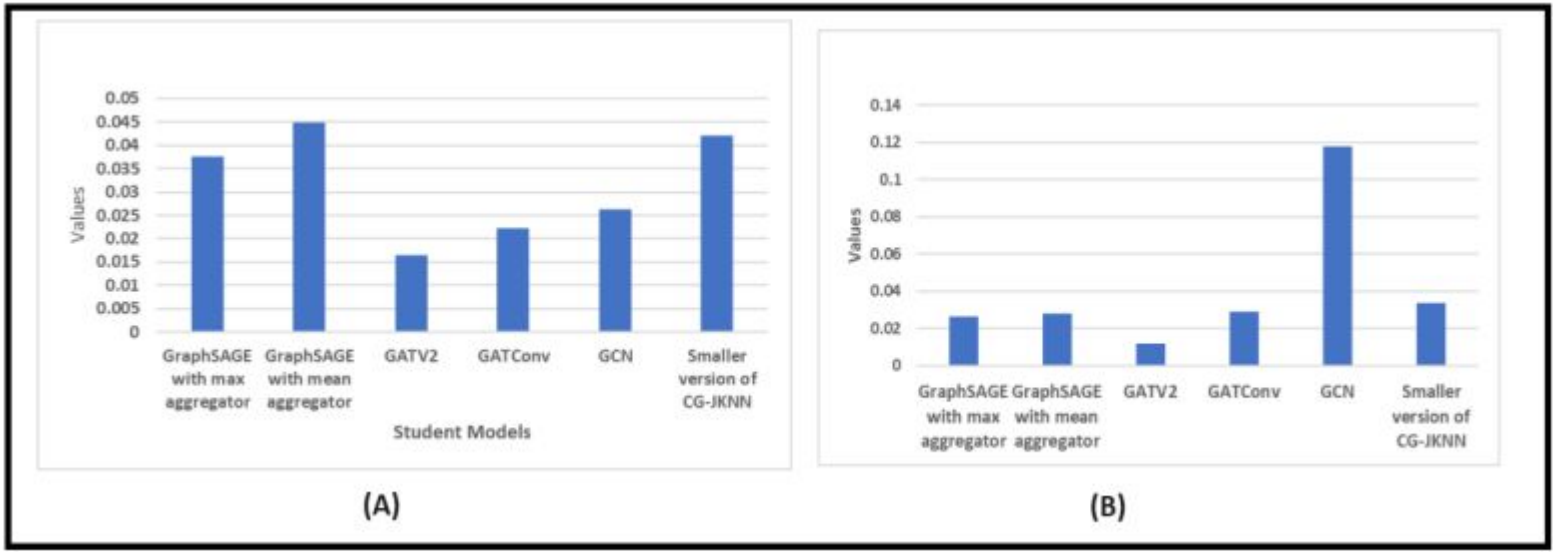


Figure 8: Distillation quality score attained by student models while using test F1 score. (B). Distillation Loss attained by student models while using test AUCPR

Best AUCPR of models with and without Distillation

Table : AUCPR Comparison

Model	Test AUCPR without Distillation	Test AUCPR with distillation
GraphSAGE_mean	0.9646	0.9822
GraphSAGE_max	**0.9753**	**0.9897**
GATV2	**0.9709**	**0.9864**
GatConv	0.9484	0.9655
GCN	0.8592	0.8801
Smaller version of CG-JKNN	–	0.9883

*Note: – indicates no value was recorded.

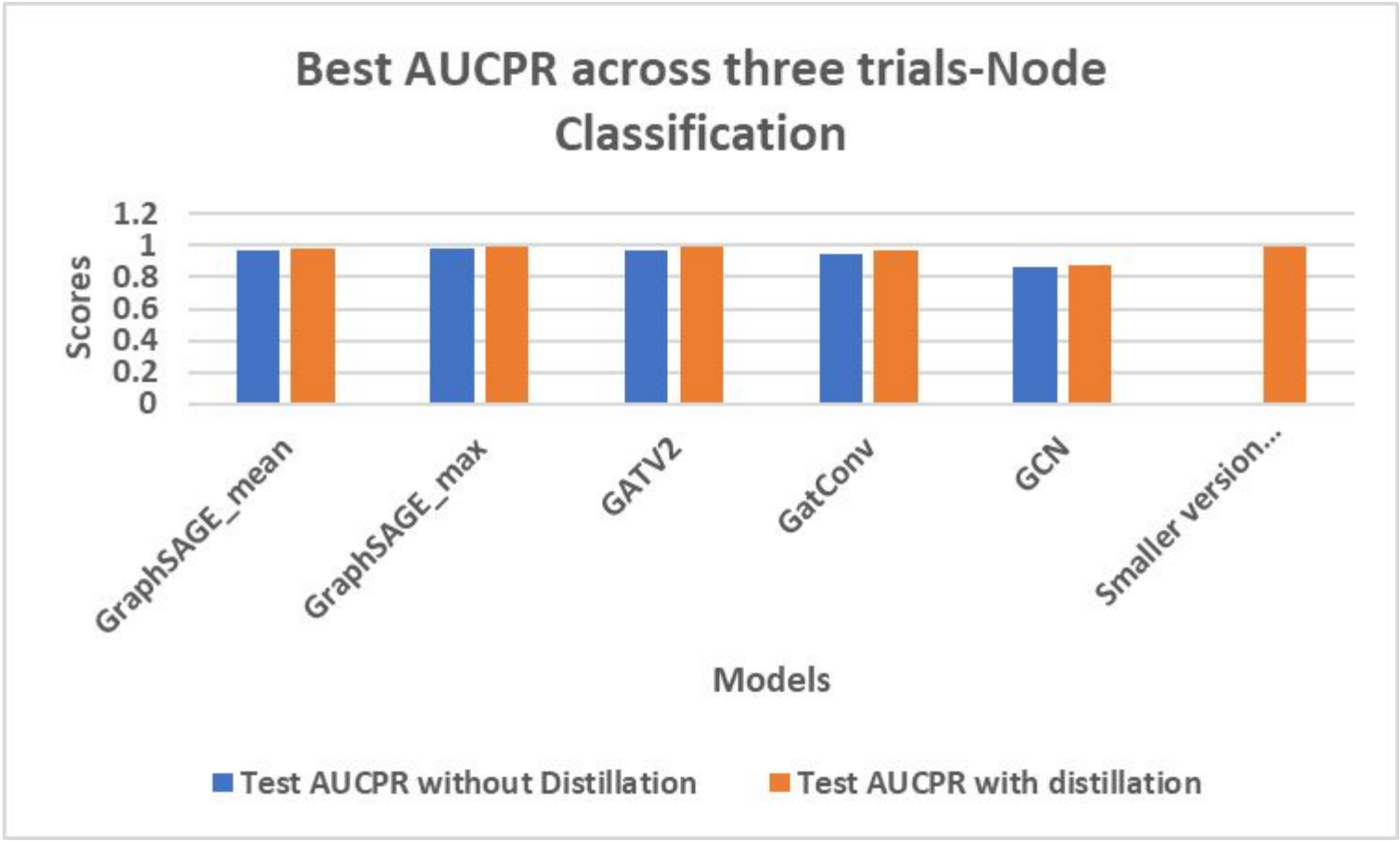


Figure: Best AUCPR

Graph Classification Results

Table 7: Combined Features: Graph Classification Performance and Hyperparameters of Grpah models with Knowledge Distillation

Model	Train_acc	Val_Acc	Test_Acc	Train_F1	Val_F1	Test_F1	Hyperparameters
Teacher Model	74.25 ± 0.52	78.26 ± 0.9	80.73 ± 0.56	0.7425 ± 0.0052	0.7826 ± 0.0090	0.8073 ± 0.0056	learning_rate: 0.0001, dropout: 0.125, graph_dropout: 0.1, num_layers: 16, hidden_channels: 32
GIN	67.37 ± 5.28	69.39 ±5.19	67.53 ±8.61	0.6737 ±0.0528	0.6939 ±0.0519	0.6753 ±0.0861	learning_rate: 0.06038, dropout: 0.38788, num_layers: 3, hidden_channels: 16
GCN	**73.53 ±0.41**	**79.58 ±0.35**	**80.73 ±0.37**	**0.7353 ±0.0041**	**0.7958 ±0.0035**	**0.8073 ±0.0037**	learning_rate: 0.00022, dropout: 0.0149, num_layers: 6, hidden_channels: 32
GATv2	73.50 ±0.43	77.19 ±1.63	79.86 ±0.44	0.7350 ±0.0043	0.7719 ±0.0163	0.7986 ±0.0044	learning_rate: 0.00074, dropout: 0.3527, num_layers: 2, hidden_channels: 16, num_heads: 4
GraphSAGE	74.64 ±0.46	76.25 ±2.58	78.39 ±0.21	0.7464 ±0.0046	0.7625 ±0.0258	0.7839 ±0.0021	learning_rate: 0.00713, dropout: 0.33502, num_layers: 3, hidden_channels: 64
CG-JKNN (smaller model)	74.91 ± 0.36	72.72 ± 2.28	73 ± 0.12	0.7491 ± 0.0036	0.7272 ± 0.0228	0.73 ± 0.0012	learning_rate: 0.0191921, dropout: 0.232173, graph_dropout: 0.0113597, num_layers: 2, hidden_channels: 32

Table 9: The Impact of Knowledge Distillation on Model Performance: Graph Classification

Model	Best Test F1 without Knowledge Distillation	Best Test F1 with Knowledge Distillation	Improvement (%)	Best Test AUCPR without Knowledge Distillation	Best AUCPR with Knowledge Distillation	Improvement (%)
GIN	0.7413	0.75	1.17	0.7728	0.8059	4.28
GCN	**0.7718**	**0.8125**	**5.27**	**0.7785**	**0.819**	**4.33**
GATv2	0.7648	0.8047	5.22	0.7437	0.7697	3.5
GraphSAGE	0.7651	0.7865	2.8	0.7978	0.8352	4.69

Observations

- **F1 Score Enhancements:** Knowledge distillation has led to improvements in the Best Test F1 score across all models, indicating enhanced model accuracy. The GCN model shows the most significant improvement of over 5%.
- **AUCPR Improvements**: There is also a noticeable improvement in the Best Test AUCPR, which measures the model's ability to distinguish between classes. The GraphSAGE model exhibits the highest increase in performance, with over a 4% improvement.
- **Effectiveness of Knowledge Distillation**: The percentage improvements indicate that knowledge distillation is effective in refining the models' classification capabilities, with substantial gains in both F1 score and AUCPR.
- **Model-Specific Insights**: The varying degrees of improvement across different models suggest that knowledge distillation’s effectiveness might be model-specific, potentially influenced by the inherent architectural features of each model.

Table 11: Comparison of Model Complexity and Distillation Quality using F1 and AUCPR Scores: Graph Classification. Model shown in bold is best performing

Model	Number_of_Parameters	Test AUCPR	Test F1 Score	Distillation Quality with F1 Score	Distillation Quality with AUCPR score
GIN	2258	0.8059	0.75	0.057	0.035
GCN	**4802**	**0.819**	**0.8125**	**0.039**	**0.048**
GATv2	4322	0.7697	0.8047	0.04	0.073
GraphSAGE	10370	0.8352	0.7865	0.1	0.083
Smaller version of CG-JKNN	4324	0.6896	0.7292	0.0487	0.0388

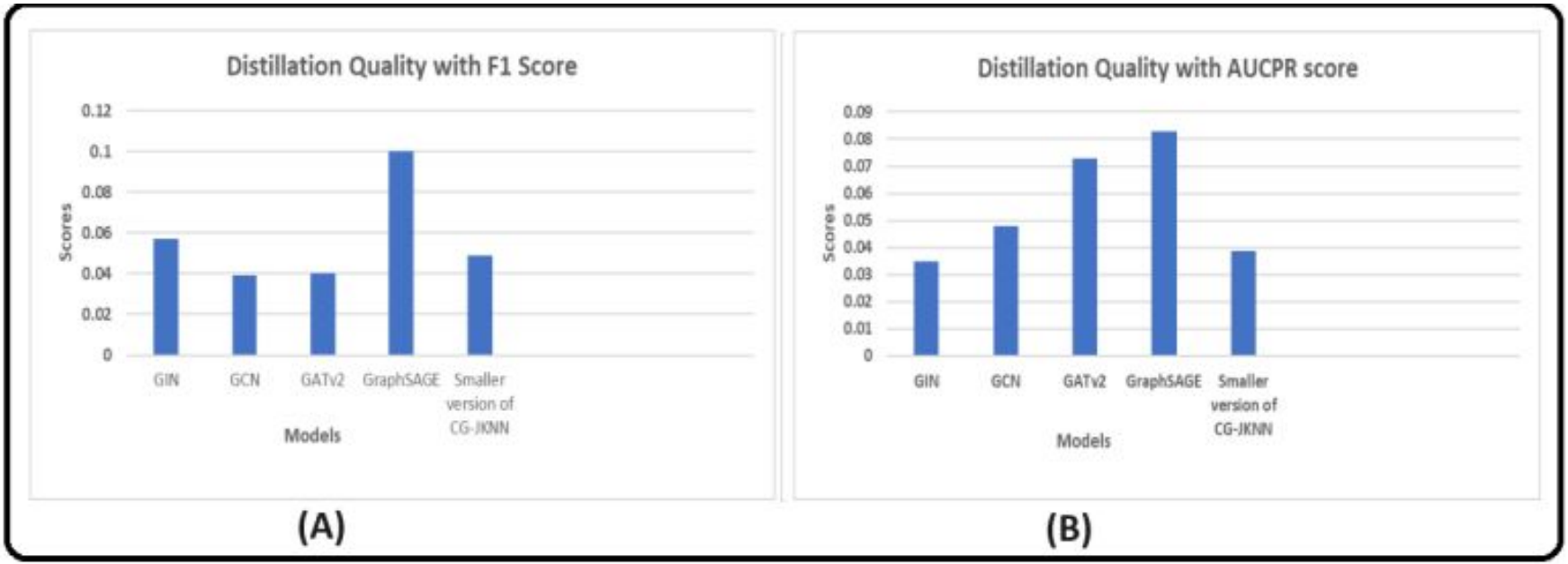


Figure : Distillation quality score attained by student models while using test F1 score. (B). Distillation Loss attained by student models while using test AUCPR.

Table 12: Comparison with state of the art models that utilize proteins dataset

Models	**Test Accuracy**	**Test F1**	Test AUCPR	**Reference**
GIN ϵ	72.2 ± 0.6	—	—	[47]
GIN-ϵ-JK	72.2 ± 0.7	—	—	[48]
DGCL+GIN	73.8 ± 3.9	—	—	[49]
GIN	73.5 ± 3.8	—	—	[50]
DGCNN	72.9 ± 3.5	—	—	[51]
CurGraph + GIN	74.7 ± 3.7	—	—	[52]
CurGraph + EigenPool	75.4 ± 3.1	—	—	[52]
Distilled GCN	$\mathbf{80.7 \pm 0.4}$	0.81 ± 0.004	0.82 ± 0.003	-
Distilled GCN with Adaptive Temperature Scaling	73.96 ± 0.52	0.7396 ± 0.0052	0.7074 ± 0.0066	-

Best AUCPR of models with and without Distillation

Table : AUCPR Comparison

Model	Test AUCPR without Distillation	Test AUCPR with distillation
GIN	0.7379	0.8059
GCN	**0.7071**	**0.819**
GATv2	0.6824	0.7697
GraphSAGE	0.7299	0.8352
Smaller version of CG-JKNN	0	0.6896

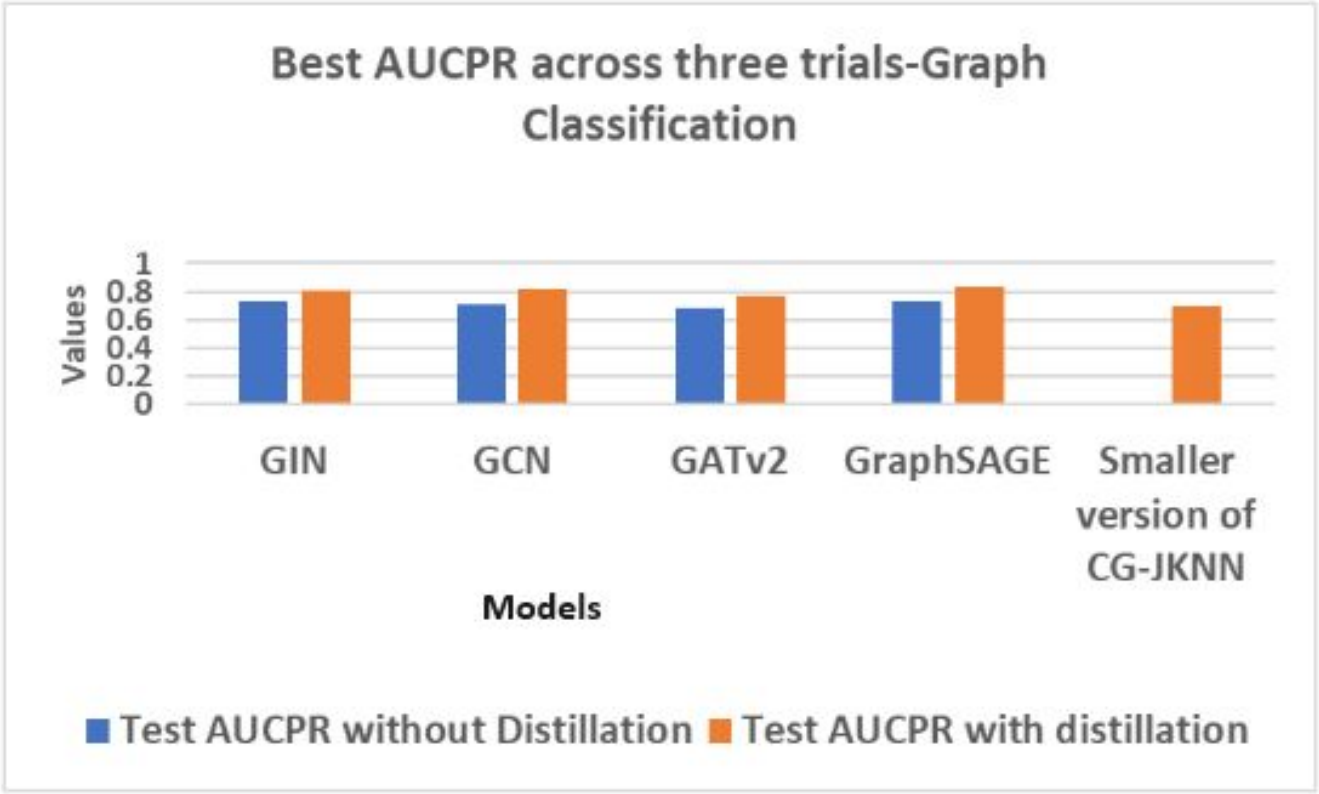


Figure: Best AUCPR

Ablation Studies

1. In this first ablation study, to address and minimize sampling bias, we conducted three distinct trials, with each trial being trained and validated on separate data subsets. This strategy ensures that each trial is exposed to different aspects of the dataset, thereby enhancing the robustness and generalizability of our findings.
2. In the second ablation study, we were motivated to incorporate adaptive temperature scaling into our knowledge distillation process by Hinton's explanation of the idea of "dark knowledge".
3. In our experimental setup, we controlled for external factors that could potentially influence the outcomes, particularly focusing on the impact of adaptive temperature scaling.
4. To do this we did not use random weight initialization for our models.
5. Furthermore, to eliminate the variability that might arise from data sampling, we maintained a uniform dataset across all three trials.

Adaptive Temperature Scaling: Approach

1. Temperature Scheduler is implemented in order to achieve this adaptation. It is intended to modify the temperature linearly throughout training.
2. Over the course of 50 epochs, the scheduler is initialised with an initial temperature of 8.0 and a final temperature of 5.0.
3. With this configuration, as the training goes on, the temperature can drop progressively from 8.0 to 5.0.
4. This adaptive strategy is justified by beginning the training process at a higher temperature, which initially softens the output distribution of the teacher's model more.
5. To further integrate this adaptive temperature mechanism into the distillation process, we modify the knowledge distillation loss function to dynamically include the current temperature value obtained from the Temperature Scheduler. Rest of the process remains the same.

$$T_{\text{current}} = T_{\text{initial}} + \frac{(T_{\text{final}} - T_{\text{initial}}) \times \text{epoch}}{\text{total\_epochs}}$$

Node Classification Results

Table 13: Student Model performance with Adaptive Temperature Scaling:Node Classification. Model shown in bold is best performing

Model	Train Acc (%)	Val Acc (%)	Test Acc (%)	Train F1	Val F1	Test F1
GraphSAGE_mean	93.34 ± 0.03	90.01 ± 0.06	87.21 ± 0.05	0.9612 ± 0.0005	0.9527 ± 0.0000	0.936 ± 0.002
GraphSAGE_max	91.54 ± 0.44	89.82 ± 0.32	82.58 ± 0.74	0.9624 ± 0.0016	0.9676 ± 0.0019	0.9434 ± 0.0008
GATV2	92.98 ± 0.74	90.24 ± 0.47	84.51 ± 0.48	0.9565 ± 0.0016	0.9465 ± 0.0036	0.9448 ± 0.0048
GatConv	90.68 ± 0.03	88.83 ± 0.35	83.39 ± 0.26	0.9544 ± 0.0093	0.9402 ± 0.0209	0.8972 ± 0.008
GCN	49.86 ± 0.58	46.75 ± 2.09	61.08 ± 1.53	0.8774 ± 0.0664	0.7816 ± 0.0468	0.8545 ± 0.0169
CG-JKNN (smaller Student Model)	92.99 ± 0.175	90.23 ± 0.402	86.24 ± 1.107	0.9594 ± 0.0023	0.9583 ± 0.003	0.9346 ± 0.0098

Table 14: Student Model performance comparison with and without Adaptive Temperature Scaling (ATS): Node Classification. KD denotes knowledge distillation.

Model	Best Test F1 with KD	Best Test F1 (% Change)		Best Test AUCPR with KD	Best Test AUCPR (% Change)	
		with KD+ATS	Increase/Decrease		with KD+ATS	Increase/Decrease
GraphSAGE_mean	0.93413	0.938	0.414 ↑	0.9882	0.9855	0.273 ↓
GraphSAGE_max	0.9484	0.9442	0.443 ↓	0.9897	0.98395	0.581 ↓
GATV2	0.9539	0.9496	0.451 ↓	0.9864	0.97672	0.981 ↓
GatConv	0.9342	0.9052	3.104 ↓	0.9655	0.9307	3.604 ↓
GCN	0.9326	0.8714	6.562 ↓	0.8801	0.89068	1.202 ↑
CG-JKNN	0.9542	0.9444	1.027 ↓	0.9883	0.9859	0.243 ↓

Table 15: Combined Features: Node Classification Performance of Graph models with Knowledge Distillation and different data subset in each trial. Model shown in bold is best performing

Model	Train_Acc	Val_Acc	Test_Acc	Train_F1	Val_F1	Test_F1
Teacher Model (CG-JKNN)	95.69 ±1.92	91.33 ±0.72	86.15 ±1.56	0.97 ±0.01	0.96 ±0.01	0.97 ±0.01
GraphSAGE_mean	91.75 ±0.93	89.44 ±0.69	85.87 ±0.99	0.93 ±0.02	0.94 ±0.02	0.92 ±0.01
GraphSAGE_max	91.40 ±0.91	89.58 ±1.43	81.38 ±2.39	0.94 ±0.02	0.93 ±0.03	0.93 ±0.01
GATV2	92.35 ±0.12	89.93 ±0.18	86.67 ±0.93	0.96 ±0.00	0.96 ±0.01	0.93 ±0.02
GatConv	88.36 ±1.37	84.46 ±3.31	78.39 ±3.48	0.94 ±0.02	0.93 ±0.03	0.90 ±0.02
GCN	77.02 ±18.55	75.01 ±17.64	76.01 ±10.85	0.94 ±0.03	0.93 ±0.02	0.90 ±0.02
CG-JKNN (smaller Student Model)	91.52 ±0.75	89.06 ±1.11	84.26 ±1.73	0.94 ±0.02	0.94 ±0.03	0.92 ±0.03

Observations

- **Influence of ATS on F1 Score**: ATS has led to changes in the Best Test F1 scores, with models like GraphSAGE_mean showing improvements, whereas others like GCN, GATv2 and GATConv demonstrate a slight decrease in performance.
- **AUCPR Score Adjustments**: GCN shows an increase in AUCPR, suggesting that ATS helps these models in distinguishing between classes more effectively.
- While ATS can improve performance in some cases, it may not universally benefit all models or may require further tuning to do so.
- The performance metrics of graph models across different data subsets show minor variations, indicating that changes in the data subset for each trial do not significantly impact the results.

Graph Classification Results

Table 14: Student Model performance with Adaptive Temperature Scaling: Graph Classification. Model shown in bold is best performing

Model	Train Acc (%)	Val Acc (%)	Test Acc (%)	Train F1	Val F1	Test F1
GIN	65.70 ±5.62	67.25 ±11.46	65.76 ±5.08	0.6570 ±0.0562	0.6725 ±0.1146	0.6576 ±0.0508
GCN	74.84 ±0.27	77.33 ±0.53	73.96 ±0.52	0.7484 ±0.0027	0.7733 ±0.0053	0.7396 ±0.0052
GATv2	73.50 ±0.43	77.47 ±2.52	72.66 ±0.52	0.7350 ±0.0043	0.7747 ±0.0252	0.7266 ±0.0052
GraphSAGE	74.39 ±0.56	76.16 ±0.92	71.61 ±1.56	0.7439 ±0.0056	0.7616 ±0.0092	0.7161 ±0.0156
CG-JKNN (smaller student model)	73.11 ±0.95	77.08 ±1.06	71.09 ±0.0	0.7311 ±0.0095	0.7708 ±0.0106	0.7109 ±0.0

Table 17: Student Model performance comparison with and without Adaptive Temperature Scaling (ATS): Graph Classification. KD denotes knowledge distillation.

Model	Best Test F1 with KD	Best Test F1 with KD+ATS	% Change	Best Test AUCPR with KD	Best Test AUCPR with KD+ATS	% Change
GIN	0.75	0.7083	5.56↓	0.8059	0.7062	12.371↓
GCN	0.8125	0.7448	8.332↓	0.819	0.7139	12.833↓
GATv2	0.8047	0.7318	9.059↓	0.7697	0.7171	6.834↓
GraphSAGE	0.7865	0.7318	6.955↓	0.8352	0.7021	15.936↓
CG-JKNN	0.7292	0.7109	2.51↓	0.6896	0.732	6.148↑

Table 20: Combined Features: Graph Classification Performance of Graph models with Knowledge Distillation and different data subset in each trial. Model shown in bold is best performing

Model	Train_Acc	Val_Acc	Test_Acc	Train_F1	Val_F1	Test_F1
Teacher Model (CG-JKNN)	75.43±0.47	73.06±4.44	76.3±0.64	0.754±0.005	0.731±0.044	0.763±0.006
GIN	64.53±6.27	62.43±5.05	61.55±4.26	0.645±0.063	0.624±0.05	0.615±0.043
GCN	74.63±0.12	72.77±4.39	73.18±2.88	0.746±0.001	0.728±0.044	0.732±0.029
GATv2	75.03±0.6	72.61±2.75	74.31±1	0.75±0.006	0.726±0.028	0.743±0.01
GraphSAGE	75.66±0.26	73.58±1.39	75.17±1.54	0.757±0.003	0.736±0.014	0.752±0.015
CG-JKNN (smaller student model)	75.4±0.73	73.22±4.32	76.13±1.73	0.754±0.007	0.732±0.043	0.761±0.017

Observations

- The table indicates that the Best Test AUCPR improves from 0.6896 with KD to 0.732 with KD+ATS, which corresponds to a 6.148% increase when considering CG-JKNN model
- For the CG-JKNN model, ATS has a beneficial impact on the AUCPR metric, contrary to the impact on the other models or metrics shown in the table.
- We believe that the effect of ATS is model-dependent, and that it could be more effective for certain models or under specific conditions.
- Teacher model (CG-JKNN) exhibits a lack of robustness to changes in the dataset by highlighting the decrease in performance metrics.
- The teacher model (CG-JKNN) experienced a noticeable decrease in its Test F1 score, from 0.8129 to 0.769, following a change in the data subset (a decline of approximately 5.4%.)
- Similarly, the best performing GCN also saw a decline of 6.16%.

Pearson Correlation: Node Classification

- One statistical tool for assessing the linear relationship between two variables is the Pearson correlation coefficient.
- The F1 score and the AUCPR (Area Under the Curve of the Precision-Recall curve) are two distinct measures that we have used to investigate the link between the number of parameters in different GNN models and their associated distillation quality score in this context.
- **A substantial positive relationship with a value of 0.9153 was discovered when we calculated the correlation using the distillation quality score F1 score.**
- **When the distillation quality score is calculated using the F1 score, this high value shows that there is a strong linear relationship between the number of parameters and the quality. This means that, in practise, models with more parameters typically show bigger distillation quality score when evaluated with the F1 score, which may imply that more complex models have a harder time distilling knowledge from the teacher model using this metric.**

- Conversely, the correlation value was around -0.3120 when we utilised the AUCPR score for distillation quality score.
- When utilising the AUCPR score as the measurement, this negative value suggests a weak and inverse association between the number of parameters and the distillation quality score, indicating that an increase in the number of parameters does not consistently result in an increase in distillation quality score.
- The different outcomes from these two measurements highlight the fact that our understanding of how effectively a student model may learn from a teacher model can be greatly impacted by the metric used for assessing distillation quality.

Pearson Correlation: Graph Classification

- For the F1 score, a correlation of approximately 0.8653 suggests that models with more parameters tend to have higher (thus worse) distillation quality scores related to F1.
- For the AUCPR score, a correlation of approximately 0.789 indicates a similar trend; more parameters are associated with higher (worse) distillation quality scores.
- These correlations would imply that more complex models (with more parameters) are not distilling or generalizing their knowledge as efficiently as simpler models when it comes to the task in hand.

Conclusion

- Confirmed that knowledge distillation significantly improves the performance of graph neural network models.
- The comprehensive set of features, combining graph and morphological attributes, proved to be highly effective for our models while performing the node level classification.
- The inclusion of global graph features with existing protein features (helices, sheets, and turns) resulted in improved model performance.
- Comparative analysis demonstrated that our approach outperforms state-of-the-art methods, underscoring the value of the integrated features.
- Pearson correlation provided valuable insights into the interplay between model complexity and efficacy.
- The selected features hold biological significance, ensuring the model's relevance to practical biological contexts.

Future Scope

- Propose a distillation method using information extracted from NNs for non-NN models [16].
- Experiment with CNN extracted features.
- Use a local structure preserving based KD method to transfer the knowledge effectively between different GNN models [17].
- Come up with a distillation score metric that is generalizable across GNNs and traditional machine learning models.
- Utilize Tissue graph along with Cell graph and measure the change in the performance.
- Employ Knowledge Distillation with Feature Maps for cell graph GNNs, which improves the effectiveness of KD by learning the feature maps from the teacher network [18].
- Establishing real-time feedback systems where the student model's outputs are continuously compared with those of the teacher model.

References

[1] G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” arXiv preprint arXiv:1503.02531, 2015.

[2] D. Koyuncu, M. K. K. Niazi, T. Tavolara, C. Abeijon, M. L. Ginese, Y. Liao, C. Mark, A. Specht, A. C. Gower, B. I. Restrepo et al., “Cxcl1: A new diagnostic biomarker for human tuberculosis discovered using diversity outbred mice,” PLoS Pathogens, vol. 17, no. 8, p.e1009773, 2021.

[3] C. Morris, N. M. Kriege, F. Bause, K. Kersting, P. Mutzel, and M. Neumann, “Tudataset: A collection of benchmark datasets for learning with graphs,” arXiv preprint arXiv:2007.08663 , 2020

[4].Yuan, Li, et al. "Revisiting knowledge distillation via label smoothing regularization." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020.

[5].] L. Studer, J. Wallau, H. Dawson, I. Zlobec, and A. Fischer, “Classification of intestinal gland cell-graphs using graph neural networks,” in 2020 25th International conference on pattern recognition (ICPR). IEEE, 2021, pp. 3636–3643.

[6]. C. Warrender, S. Forrest, and F. Koster, “Modeling intercellular interactions in early mycobacterium infection,” Bulletin of mathematical biology, vol. 68, pp. 2233–2261, 2006.

[7] V. Chadalapaka, V. Ustun, and L. Liu, “Leveraging graph networks to model environments in reinforcement learning,” in The International FLAIRS Conference Proceedings, vol. 36, 2023.

[8] Z. Guo, C. Zhang, Y. Fan, Y. Tian, C. Zhang, and N.V. Chawla, “Boosting Graph Neural Networks via Adaptive Knowledge Distillation”.

[9]. Pati, P., Jaume, G., Foncubierta-Rodriguez, A., Feroce, F., Anniciello, A. M., Scognamiglio, G., ... & Gabrani, M. (2022). Hierarchical graph representations in digital pathology. Medical image analysis, 75, 102264.

[10]. Y. Wang, Y. G. Wang, C. Hu, M. Li, Y. Fan, N. Otter, I. Sam, H. Gou, Y. Hu, T. Kwok et al., “Cell graph neural networks enable the precise prediction of patient survival in gastric cancer,” NPJ precision oncology, vol. 6, no. 1, pp. 1–12, 2022.

[11]. Bilgin, Cagatay, et al. "Cell-graph mining for breast tissue modeling and classification." 2007 29th Annual international conference of the IEEE Engineering in Medicine and Biology Society. IEEE, 2007.

[12] W.L. Hamilton, R. Ying, and J. Leskovec. 2017. “Inductive Representation Learning on Large Graphs.” In Advances in Neural Information Processing Systems.

[13] K. Xu, C. Li, Y. Tian, T. Sonobe, K.-i. Kawarabayashi, and S. Jegelka, “Representation learning on graphs with jumping knowledge networks,” in International conference on machine learning. PMLR, 2018, pp. 5453–5462.

[14] J. Kim, J. Jung, and U. Kang, “Compressing deep graph convolution network with multi-staged knowledge distillation,” Plos one, vol. 16, no. 8, p. e0256187, 2021.

[15] Brody S, Alon U, Yahav E. How attentive are graph attention networks? arXiv preprint arXiv:2105.14491. 2021.

[16]. Fukui, Shota, Jaehoon Yu, and Masanori Hashimoto. "Distilling knowledge for non-neural networks." *2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2019.

[17]. Yang, Y.; Qiu, J.; Song, M.; Tao, D.; and Wang, X. 2020. Distilling knowledge from graph convolutional networks. In *CVPR*.

[18]. Chen, Wei-Chun, Chia-Che Chang, and Che-Rung Lee. "Knowledge distillation with feature maps for image classification." *Computer Vision–ACCV 2018: 14th Asian Conference on Computer Vision, Perth, Australia, December 2–6, 2018, Revised Selected Papers, Part III 14*. Springer International Publishing, 2019.

[19]. Guo, Zhichun, et al. "Boosting graph neural networks via adaptive knowledge distillation." *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 37. No. 6. 2023.

[20] Y. Jing, Y. Yang, X. Wang, M. Song, and D. Tao, “Amalgamating knowledge from heterogeneous graph neural networks,” in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[21] S. Antaris and D. Rafailidis, “Distill2vec: Dynamic graph representation learning with knowledge distillation,” *CoRR*, vol. abs/2011.05664, 2020.

[22] B. Yan, C. Wang, G. Guo, and Y. Lou, “Tinygnn: Learning efficient graph neural networks,” in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ser. KDD ’20.

[23]. V. Deshmukh, S. Baskar, T. Berger, E. Bradley, and J. Meiss, “Comparing feature sets and machine-learning models for prediction of solar flares-topology, physics, and model complexity,” Astronomy & Astrophysics, vol. 674, p. A159, 202