



# Edge-Based Graph Neural Networks for Cell-Graph Modeling and Prediction

Tai Hasegawa<sup>1</sup>, Helena Arvidsson<sup>2</sup>, Nikolce Tudzarovski<sup>2</sup>,  
Karl Meinke<sup>1</sup>, Rachael V. Sugars<sup>2</sup>, and Aravind Ashok Nair<sup>1</sup>

<sup>1</sup> KTH Royal Institute of Technology, Stockholm, Sweden  
hasegawa.t.as@m.titech.ac.jp, {karlm,aanair}@kth.se

<sup>2</sup> Karolinska Institutet, Stockholm, Sweden  
{helena.arvidsson,nikolce.tudzarovski,rachael.sugars}@ki.se

**Abstract.** Identification and classification of cell-graph features using graph-neural networks (GNNs) has been shown to be useful in digital pathology. In this work, we consider the role of edge labels in cell-graph modeling, including histological modeling techniques, edge aggregation in GNN architectures, and edge label prediction. We propose EAGNN (Edge Aggregated GNN), a new GNN model that aggregates both node and edge label information to take advantage of topological information about cellular data and facilitate edge label prediction. We introduce new edge label features that improve histological modeling and prediction. We evaluate our EAGNN model for the task of detecting the presence and location of the basement membrane in oral mucosal tissue, as a proof-of-concept application.

**Keywords:** Digital Pathology · Graph Neural Network · Cell-Graph · Basement Membrane · Oral Mucosa

## 1 Introduction

To capture intra-cellular and high-level histological relationships in whole-slide images (WSI) of tissue samples, cell-graph models have been considered [13]. In a cell-graph, properties of cells and interactions between them are represented by labelled nodes and edges. Graph neural networks (GNNs) are a specific class of machine learning (ML) algorithm which have been applied to cell-graph models to locate and classify complex histological features [4, 16, 20]. In this work, we consider the role of edge labels in cell-graph modeling and GNN-based model analysis including: (i) histological modeling using edge labels, (ii) edge aggregation in GNNs, and (iii) edge label classification algorithms.

Our study will mainly focus on new GNN algorithms for aggregating node and edge data, with a view to making edge label predictions in a cell-graph model. We also propose new types of edge features, going beyond the simple geometric distance label found in the literature. These new edge labels are shown to be effective for improved histological analysis by means of ML ablation studies.

We evaluate the new GNN models and edge features on a representative digital pathology task of predicting the presence and location of the basement membrane (BM) in hematoxylin and eosin (H&E) stained oral mucosa samples. Structural properties of the BM play an important role in classifying oral diseases such as chronic graft-versus-host disease (oral cGvHD) [23].

## 1.1 Contributions of This Work

The main contributions of this work are as follows:

1. We propose EAGNN as a novel message passing model<sup>1</sup> for predicting cell-graph properties. EAGNN aggregates both node and edge label data to yield edge label predictions.
2. We propose two new types of edge classification algorithm, which can be used on the backend of EAGNN to make edge label predictions.
3. We propose three new edge label features for cell-graph analyses: (a) cell density difference, (b) cell entropy difference, and (c) neighborhood overlap similarity.
4. We evaluate different combinations of EAGNN with edge classifiers and edge label features for the prediction of BM location in oral mucosa images. We show that EAGNN can significantly outperform simple node-based aggregation across a wide variety of performance measures.

The organisation of this work is as follows. In Sect. 2 we consider background and related research on GNNs for learning cell-graph models. In Sect. 3 we discuss message passing GNNs and graph data aggregation. In Sect. 4 we define the EAGNN model with edge label aggregation and edge classification algorithms. In Sect. 5 we evaluate EAGNN on the task of predicting BM integrity in healthy and diseased oral tissue images and compare it with simple node-based aggregation. In Sect. 6 we discuss some limitations of this study and address the possible directions to overcome them. Finally, in Sect. 7 we present some conclusions.

## 2 Background and Related Work

Mapping the structural features of tissues is key to understanding and making a diagnosis of disease severity in digital pathology, for example the BM. An accurate estimation of the BM location and integrity is an important feature in health and disease, but a challenging task. Several studies have considered this problem [6, 25, 26]. However, most of these methods depend upon pixel level information that fails to capture histological and topological relationships between the BM and various cells locally present in the tissue. In this study, we used cell-graphs and graph-based ML techniques to map oral mucosal tissue and locate the BM in healthy and diseased situations.

---

<sup>1</sup> Official PyTorch implementation of the EAGNN algorithm is publicly available at <https://github.com/aravi11/EAGNN>.

Generally in the literature on digital pathology, node labels alone have been used for cell-graph modeling and GNN training [1]. To our knowledge, [17] is the only work that has considered the use of cellular interactions in a cell graph for identifying the BM. However the GNN model used in [17] does not utilize the topological information encoded in graph edges during the learning process. Other GNN case studies also utilize edge features such as distance [2, 5, 21, 22] or edge weights [9] for digital pathology. However, in all these approaches the edge features are simple one-dimensional real-valued features. The use of multi-dimensional edge features for optimal cell-graph representation and prediction is currently heavily under-explored.

### 3 Graph Neural Networks for Cell-Graph Learning

#### 3.1 Cell-Graphs as Labelled Graph Structures

A cell-graph is a mathematical model of histological tissue features that can represent nuclei and the interactions between nuclei. This model is motivated by the hypothesis that cells in a tissue organize to perform a specific function [27].

Formally, a cell-graph:  $G = (\mathcal{V}, \mathcal{E}, l_{\mathcal{V}} : \mathcal{V} \rightarrow \mathbb{R}^D, l_{\mathcal{E}} : \mathcal{E} \rightarrow \mathbb{R}^P)$ , is a labelled undirected graph where  $\mathcal{V} = \{v_1, \dots, v_n\}$  is a finite set of nodes, and  $n$  is the size of the graph. Furthermore,  $\mathcal{E} = \{e_1, \dots, e_m\}$  is a finite set of edges, and each edge  $e \in \mathcal{E}$  is an unordered pair of nodes  $e = \{v_i, v_j\}$ . A node  $v_j$  is termed an immediate neighbour of  $v_i$  if there exists an edge  $\{v_j, v_i\} \in \mathcal{E}$ . We let  $\mathcal{N}(v)$  denote the set of all immediate neighbours of  $v$ . The degree of a node  $v$  is the size of its immediate neighbour set,  $deg(v) = |\mathcal{N}(v)|$ . The functions  $l_{\mathcal{V}}$  and  $l_{\mathcal{E}}$  are node and edge labelling functions of  $D$  and  $P$  features respectively.

To apply methods of linear algebra to graph learning problems, an edge set  $\mathcal{E}$  can be encoded by an adjacency matrix  $\mathbf{A} \in \mathbb{R}^{n \times n}$  of real values, where  $n$  is the graph size. The matrix values  $\mathbf{A}_{uv}$  and  $\mathbf{A}_{vu}$  are both set to 1.0 if there exists an edge between node  $u$  and node  $v$  in  $\mathcal{E}$ , otherwise both are set to 0.0. Moreover, the node labelling  $l_{\mathcal{V}}$  can be encoded as a node feature matrix  $\mathbf{X} \in \mathbb{R}^{n \times D}$  in which case the matrix row  $\mathbf{X}_v \in \mathbb{R}^D$  represents the feature vector for node  $v$ . Similarly, the edge labelling  $l_{\mathcal{E}}$  can be encoded as an edge feature tensor  $\mathbf{E} \in \mathbb{R}^{n \times n \times P}$ . For each pair of connected nodes  $\{u, v\} \in E$  the entry  $\mathbf{E}_{uv} \in \mathbb{R}^P$  represents the  $P$ -dimensional feature vector of the edge between node  $u$  and node  $v$ . As a simplifying notation, for any edge feature  $1 \leq p \leq P$  the matrix  $\mathbf{E}_p \in \mathbb{R}^{n \times n}$  denotes the projection of  $\mathbf{E}$  onto the single edge feature  $p$ .

#### 3.2 Graph Neural Networks

A general spatial-based GNN has a layered architecture, where at each layer  $k$ , a low-dimensional ( $d_k$ -dimensional) representation  $\mathbf{h}_u^k \in \mathbb{R}^{d_k}$  of the graph structure around node  $u$  is computed. Computation at each layer  $k$  normally consists of two stages. Firstly, for each node  $u$ , an *AGGREGATE* <sup>$k$</sup>  operation produces an

integrated representation  $\mathbf{h}_{\mathcal{N}(u)}^k$  of all immediate neighbors  $v \in \mathcal{N}(u)$  of  $u$  using the representations  $\mathbf{h}_v^{k-1}$  from layer  $k-1$  and is represented as,

$$\mathbf{h}_{\mathcal{N}(u)}^k = \text{AGGREGATE}^k(\{\mathbf{h}_v^{k-1}, \forall v \in \mathcal{N}(u)\}). \quad (1)$$

Secondly, for each node  $u$ , a *COMBINE* <sup>$k$</sup>  operation updates the representation  $\mathbf{h}_u^k$  of  $u$  by combining its previous representation  $\mathbf{h}_u^{k-1}$  on layer  $k-1$  with the aggregated representation  $\mathbf{h}_{\mathcal{N}(u)}^k$  of all its immediate neighbours  $\mathcal{N}(u)$ , using a nonlinear function,  $\mathbf{h}_u^k = \text{COMBINE}^k(\mathbf{h}_u^{k-1}, \mathbf{h}_{\mathcal{N}(u)}^k)$ .

This iterative computation over layers  $0 \leq k \leq K$  is initialized by setting  $\mathbf{h}_u^0 = \mathbf{X}_u$ . Spatial variants of GNNs [3, 10, 18] implement aggregation by matrix multiplication as:

$$\mathbf{H}_{agg}^k = \mathbf{A}\mathbf{H}^{k-1}\mathbf{W}_0^k \quad (2)$$

where  $\mathbf{H}_{agg}^k \in \mathbb{R}^{n \times d_k}$  is the tensor matrix (i.e. stack) of all aggregations  $\mathbf{h}_{\mathcal{N}(u)}^k$ ,  $\mathbf{A} \in \mathbb{R}^{n \times n}$  is the adjacency matrix,  $\mathbf{H}^{k-1} \in \mathbb{R}^{n \times d_{k-1}}$  is the tensor matrix of representations  $\mathbf{h}_v^{k-1}$  on the  $k-1$ -th layer, and  $\mathbf{W}_0^k \in \mathbb{R}^{d_{k-1} \times d_k}$  is a matrix of learnable parameters. The combine operation is formulated as:

$$\mathbf{H}^k = \sigma(\mathbf{H}_{agg}^k + \mathbf{H}^{k-1}\mathbf{W}_1^k) \quad (3)$$

where  $\mathbf{W}_1^k \in \mathbb{R}^{d_{k-1} \times d_k}$  is a second matrix of learnable parameters, and  $\sigma$  is a nonlinear function applied pointwise, such as ReLU [11]. Finally, after  $K$  layers, a low-dimensional node embedding  $\mathbf{Z} \in \mathbb{R}^{n \times d_K}$  is obtained as a tensor matrix,  $\mathbf{Z} = \mathbf{H}^K$ . A widely used spatial GNN using node aggregation alone is GraphSAGE [14].

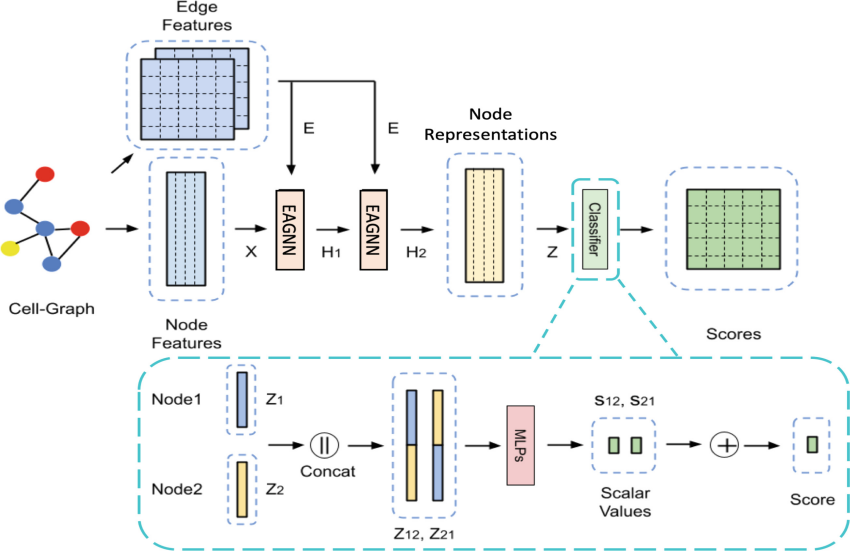
To solve edge classification problems, we need a low-dimensional edge embedding  $\mathbf{Z}_e \in \mathbb{R}^k$  of each edge  $e$ . The simplest approach to embed an edge  $e$  is to combine the final embeddings of both of its nodes.

## 4 A GNN Model for Node and Edge Aggregation

Our proposed GNN architecture is depicted in Fig. 1, and can be divided into two stages, node embedding layers and an edge classifier. The node embedding layers derive latent node representation from a cell-graph.

### 4.1 Node Embedding Layers

Inspired by EGNN(C) [12], we propose an EAGNN layer which incorporates multiple edge features to embed nodes. The major difference between the proposed EAGNN layer and EGNN(C) is the way that edge features are normalised before aggregation. This feature normalization method is explained in detail in Sect. 5.1.



**Fig. 1.** The overview of the proposed GNN architecture which consists of two EAGNN layers and a classifier. The model extracts node and edge features from a cell-graph, and outputs a score to classify each edge as a BM crossing one or not.

Following the matrix multiplication as in Eq. 2, we formulate the aggregation operation of the proposed model at layer  $k$ , named  $EAgg^k$ , as follows:

$$EAgg^k(\mathbf{E}_p, \mathbf{H}^{k-1}) = \mathbf{E}_p \mathbf{H}^{k-1} \mathbf{W}_0 \quad (4)$$

Then, we combine the previous node representation using the combine operation formulated in Sect. 3.2. We perform these aggregation and combining operations for each edge feature, and concatenate them. Therefore, the formula for the  $k$ th EAGNN layer is given by:

$$\mathbf{H}^k = \sigma \left[ \parallel_{p=1}^P (\mathbf{E}_p \mathbf{H}^{k-1} \mathbf{W}_0^{k-1} + \mathbf{H}^{k-1} \mathbf{W}_1^{k-1}) \right] \quad (5)$$

where  $\parallel$  denotes the concatenation operator. As a non-linear function  $\sigma$  we employ the ELU function [7]. Note that this non-linear function is not used in the final layer  $K$  of the node embedding layers. As described in Fig. 1, we used two EAGNN layers in our case study of BM identification: our evaluation suggests that two layers are sufficient for this task. After two embedding layers, the node representation  $\mathbf{z}_u$  is given by  $\mathbf{z}_u = \mathbf{H}_u^2$  for node  $u$ .

## 4.2 Edge Classifiers

Once the node embedding layers have computed the node representations, an edge classifier can partition all edges into two non-overlapping classes  $A, B$ . For

each edge  $e = \{u, v\} \in \mathcal{E}$ , the edge classifier computes a score  $S_{uv} \in [0, 1]$  which represents the estimated likelihood that  $e$  falls into class  $A$  or class  $B$ . The estimated likelihood score  $S_{uv}$  is compared with a class threshold criterion  $\theta$ , which represents the decision boundary between  $A$  and  $B$ .

We propose here three methods of edge classification: multiplication (MUL), negative multiplication (NegMUL) and bidirectional concatenation + multilayer perceptron (BC+MLP). The simplest classifier among the three variants is the MUL classifier. It multiplies the embedding vector of node  $u$  and node  $v$  followed by the sigmoid function:  $S_{uv} = \text{sigmoid}(\mathbf{z}_u \mathbf{z}_v^T)$ . Similarly to the MUL classifier, the NegMul classifier multiplies the embedding vectors of a pair of nodes, but unlike MUL it subtracts the output from 1,  $S_{uv} = 1 - \text{sigmoid}(\mathbf{z}_u \mathbf{z}_v^T)$ .

In contrast with the classifiers MUL and NegMUL, the BC+MLP classifier, which is depicted in Fig. 1, uses a shallow neural network approach to compute a score  $S_{uv}$  for each edge. A neural network approach to edge classification also needs to combine the embedding vectors. There are several ways to combine a pair of node representations, such as element-wise product or summation. Concatenation of node embedding vectors is a simple and effective approach, since it preserves all node information [8]. As shown in Fig. 1, the BC+MLP classifier concatenates the node embeddings  $\mathbf{z}_u, \mathbf{z}_v$  of nodes  $u, v$  in both directions ( $\mathbf{z}_{uv} = \mathbf{z}_u \parallel \mathbf{z}_v, \mathbf{z}_{vu} = \mathbf{z}_v \parallel \mathbf{z}_u$ ) to obtain a final score in an orientation-invariant way. The node embedding concatenations  $\mathbf{z}_{uv}, \mathbf{z}_{vu}$  are sequentially fed into an MLP to analyse the relationship between the nodes  $u, v$ . The scores generated from the MLP layers are combined and passed through the sigmoid activation function to obtain the final score  $S_{uv}$ <sup>2</sup> bounded in  $[0, 1]$ . A class threshold criterion  $\theta$  can be chosen as a parameter, and compared with  $S_{uv}$  to determine class membership.

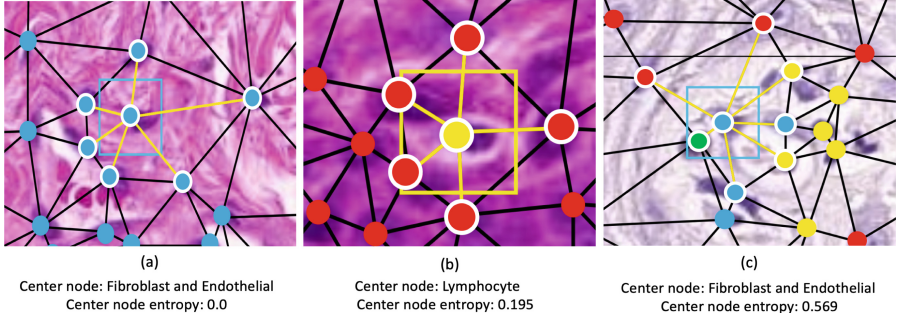
## 5 Evaluation of the EAGNN Model

In this section we compare the performance of EAGNN against the widely used GraphSage GNN which is based on node aggregation alone. The evaluation task is prediction of the BM location and integrity in cell-graph models of oral mucosa samples.

### 5.1 An Oral Mucosa Cell-Graph Dataset

For supervised training and evaluation of the EAGNN model, a dataset of ground truth cell-graphs was compiled from digitized images of H&E stained oral tissue samples. To compile the dataset, we extracted 42 tiles from WSI of oral mucosal biopsies from nine patients receiving haematopoietic cell transplantation [23]. On each tile, histology experts manually annotated the cell type and  $x, y$  centroid co-ordinates for each cell nucleus. Focusing on the cell types  $T = \{\textit{epithelial}, \textit{fibroblast and endothelial}, \textit{inflammatory}, \textit{lymphocyte}\}$ , each nucleus was then modeled by a graph node  $v_i \in V$  and labelled by its cell type  $N_{\textit{type}}(v_i) \in T$ .

<sup>2</sup> Note  $S_{uv} = S_{vu}$  and so the proposed method is invariant to node ordering.



**Fig. 2.** Examples of nodes and their entropy values. The center node is highlighted with a rectangular box. The neighboring nodes are highlighted with a white circle and yellow edges. Cell types are represented by four colours: epithelial (red), fibroblast and endothelial (blue), inflammatory (green) and lymphocyte (yellow). (Color figure online)

The location and extent of the BM in each tile was manually annotated using cubic splines. An undirected edge set  $\mathcal{E}$  for  $\mathcal{V}$  was generated using the Delaunay triangulation method and all edges  $e = \{u, v\}$  having a Euclidean length greater than 300 pixels (150 microns) were deleted<sup>3</sup>

For accurate GNN-based prediction of the BM, besides cell type  $N_{type}$ , additional node and edge labels were shown to improve the quality of BM prediction.

### Additional Node Labels

1. Cell Density: We define the (local) cell density at node  $v \in \mathcal{V}$  as the average distance between  $v$  and its immediate neighbours  $u \in \mathcal{N}(v)$ .

$$N_{den}(v) = \frac{1}{|\mathcal{N}(v)|} \sum_{u \in \mathcal{N}(v)} d(v, u). \quad (6)$$

2. Cell Entropy: For defining the (local) cell entropy of a node  $v \in \mathcal{V}$ , we first calculate the probability  $p_v(\tau)$  of finding the cell type  $\tau \in T$  in its immediate neighbourhood using Eq. 7. Cell entropy is defined using the Shannon entropy measure in Eq. 8 and visualised in Fig. 2.

$$p_v(\tau) = \frac{|\{u \in \mathcal{N}(v) \text{ or } u = v : N_{type}(u) = \tau\}|}{|\mathcal{N}(v)|} \quad (7)$$

$$N_{ent}(v) = - \sum_{\tau \in T} p_v(\tau) \times \ln p_v(\tau) \quad (8)$$

<sup>3</sup> We kept the limiting criterion equivalent to 300 pixels to avoid long edges in the cell-graph, as the cell density varies across different parts of a tile.

**Additional Edge Labels.** For GNN-based prediction of the BM, the use of edge labels significantly increased the accuracy of our model. We make use of five edge labels defined below using edge feature tensor representation (c.f. Sect. 3.1).

1. Node Distance: For each edge  $\{u, v\} \in \mathcal{E}(d)$  we use the Euclidean distance between the endpoints as the distance label:  $\hat{\mathbf{E}}_{u,v}^{\text{dis}} = d(u, v)$ .
2. Cell Density Difference: Recalling the cell density definition of Eq. 6, we can define the difference in cell density along edge  $\{u, v\} \in \mathcal{E}(d)$  by:  $\hat{\mathbf{E}}_{u,v}^{\text{den}} = N_{\text{den}}[v] - N_{\text{den}}[u]$
3. Cell Entropy Difference: Similarly to the cell density difference, we can define the cell entropy difference<sup>4</sup> along an edge  $\{u, v\} \in \mathcal{E}(d)$  by:  $\hat{\mathbf{E}}_{u,v}^{\text{ent}} = |N_{\text{ent}}[v] - N_{\text{ent}}[u]|$
4. Neighbourhood Overlap Similarity: This measure aims to quantify the overlap between the neighbourhoods of two nodes, which is useful for tasks like edge and community detection [15]. To define the (relativized) overlap between two nodes  $u$  and  $v$  we use the Sorenson similarity index defined by:  $\hat{\mathbf{E}}_{u,v}^{\text{nei}} = \frac{2|N(u) \cap N(v)|}{\text{deg}(u) + \text{deg}(v)}$
5. BM Crossing: For each edge  $\{u, v\} \in \mathcal{E}(d)$  we define the binary valued BM crossing measure:  $\hat{\mathbf{E}}_{u,v}^{\text{bm}} = 1$  if  $\{u, v\}$  crosses the BM, otherwise  $\hat{\mathbf{E}}_{u,v}^{\text{bm}} = 0$ .

**Edge Feature Normalization.** To incorporate the edge features we multiply them with the node features during the convolution operations of each EAGNN layer. In order to maintain the feature scale during the matrix multiplication, we normalize the edge feature values. There are several ways to normalize the edge feature values, including the Doubly Stochastic Normalization method proposed in [12]. However the Doubly Stochastic Normalization method assumes all the edge feature values to be non-negative which is not true in our case especially while considering the cell density difference. We normalize the edge feature values by rows as proposed in [24]. The row normalization for feature  $1 \leq p \leq P$  and edge  $(u, v) \in \mathcal{E}(d)$  is defined by:

$$\mathbf{E}_{uvp}^{\mathbf{X}} = \frac{\hat{\mathbf{E}}_{uvp}^{\mathbf{X}}}{\sum_{v' \in \mathcal{N}(u)} |\hat{\mathbf{E}}_{uv'p}^{\mathbf{X}}|}. \quad (9)$$

## 5.2 Training Setup

The cell-graph dataset was divided into two subsets: training (70%) and test data (30%). Some of the training dataset was reserved as a validation set (15%). Table 1 show the population sizes of edge classes in the data set indicating data bias to non-crossing edges. The GNN models were trained by backpropagation to minimize the mean of the binary cross entropy loss function for each mini-batch. The

<sup>4</sup> The reason for choosing the absolute difference here was to have non-negative entropy difference value. Ablation studies showed that negative cell entropy differences had an adverse effect on the efficiency of the trained model.



loss function was given by:  $\mathbf{loss} = -\frac{1}{N} \sum_{i=1}^N [\mathbf{l}_i \log(\mathbf{y}_{pred}) + (\mathbf{1} - \mathbf{l}_i) \log(\mathbf{1} - \mathbf{y}_{pred})]$ , where  $N$  is the number of samples in a mini-batch,  $l_i \in \{0, 1\}$  and  $y_{pred} \in [0, 1]$  are the label and the prediction for sample  $i$  respectively. We trained the models for 100 epochs with a batch size of 32. Each batch had 16 BM crossing edges and 16 BM non-crossing edges to avoid the bias derived from data imbalance. To optimize learnable parameters, we used Adam with an initial learning rate set to 0.001 which was scheduled to drop by 0.1 after every 40 epochs. To avoid overfitting on the training data, we used dropout with probability 0.5 for EAGNN layers and BC+MLP classifier, 0.3 for GraphSAGE, and the weight-decay parameter was set to 0.0001.

**Table 1.** Population sizes of edge classes in the oral mucosa cell-graph dataset.

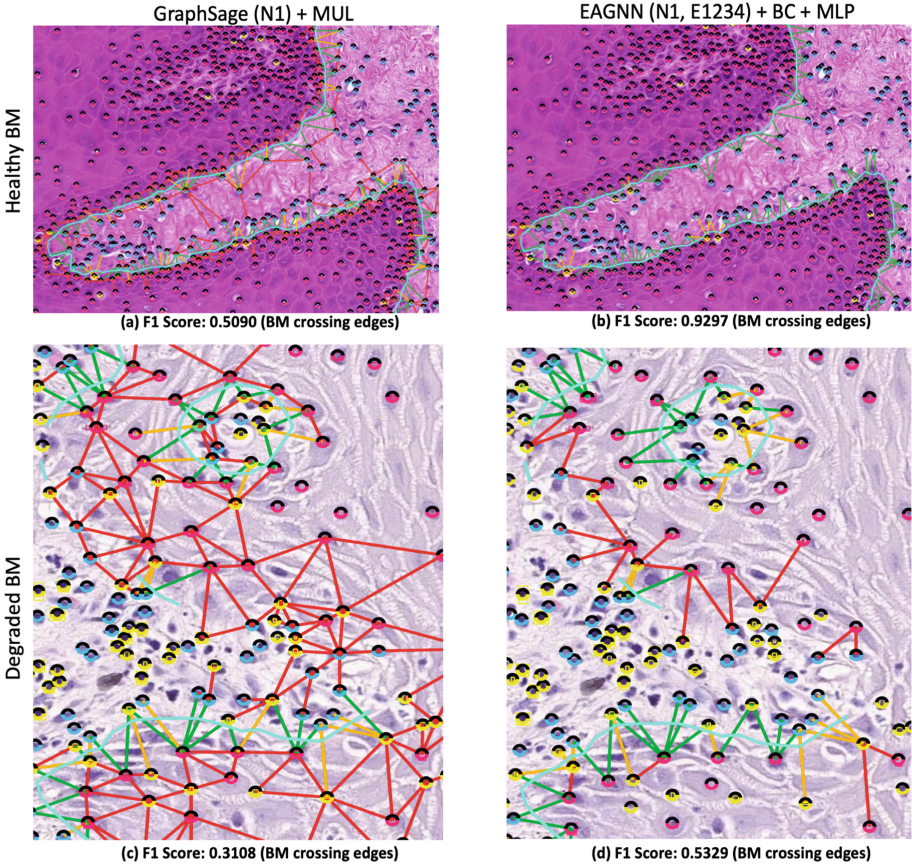
Dataset	Non-Crossing Edges	Crossing Edges	Total Edges
Training	96195	6221	102416
Validation	20017	1169	21186
Testing	45779	2733	48512
Total	161991	10123	172114

### 5.3 Discussion

To compensate for data imbalances in the ground truth dataset, we evaluated EAGNN performance on the BM prediction problem using five standard metrics: precision, recall, F1 score, ROC-AUC and accuracy.

**Table 2.** Comparison of GNN models, edge classifiers, aggregated node features (NF) and edge features (EF) for BM prediction on the oral mucosa cell-graph dataset. (NF specifies aggregated node features:  $N_1 = N_{type}$ ,  $N_2 = N_{den}$ , and  $N_3 = N_{ent}$ . EF specifies aggregated edge features:  $E_1 = \hat{E}^{dis}$ ,  $E_2 = \hat{E}^{den}$ ,  $E_3 = \hat{E}^{ent}$ , and  $E_4 = \hat{E}^{nei}$ .)

Model	Classifier	NF	EF	Precision	Recall	F1	ROC-AUC	Accuracy
GraphSAGE	MUL	$N_1$	-	0.3114	0.7494	0.4400	0.9112	0.8925
GraphSAGE	NegMUL	$N_1$	-	0.7548	0.8547	0.8016	0.9712	0.9762
GraphSAGE	BC+MLP	$N_1$	-	0.7714	<b>0.8580</b>	0.8124	0.9764	0.9777
GraphSAGE	BC+MLP	$N_{123}$	-	0.7793	0.8426	0.8097	0.9777	0.9781
EAGNN	MUL	$N_{123}$	$\hat{E}_{1234}$	0.5275	0.5258	0.5267	0.9284	0.9468
EAGNN	NegMUL	$N_{123}$	$\hat{E}_{1234}$	0.7353	0.8496	0.7883	0.9763	0.9743
EAGNN	BC+MLP	$N_{123}$	$\hat{E}_{1234}$	<b>0.8751</b>	0.7845	0.8273	<b>0.9822</b>	0.9816
EAGNN	BC+MLP	$N_1$	$\hat{E}_{1234}$	0.8548	0.8247	<b>0.8395</b>	0.9799	<b>0.9822</b>



**Fig. 3.** Comparison of EAGNN and GraphSage predictions for intact and degraded BM tissue samples. The blue line is the BM annotation. The green, yellow and red lines represent true positive, false negative and false positive edge predictions respectively. (Color figure online)

Table 2 summarizes the performance of EAGNN combined with three different types of edge classifiers in comparison with GraphSAGE [14]. The combination GraphSAGE+MUL (row 2) is the baseline architecture which was used in [17] for BM prediction. The combination of EAGNN with the BC+MLPs edge classifier outperforms the other methods in 4 out of 5 metrics. GraphSAGE with BC+MLP resulted in high recall and relatively low precision, while EAGNN with BC+MLP resulted in high precision and relatively low recall. This is caused by the selection of the class threshold criterion  $\theta$ . Tuning the parameter  $\theta$  w.r.t F1 score, helped to balance the trade-off between precision and recall. The optimum balance was  $\theta = 0.7$ .

Figure 3 shows the BM prediction results for the best and worst performing models: EAGNN+BC+MLP ( $NF = N_1, EF = E_{1234}$ ) and GraphSAGE+MUL

( $NF=N_1$ ) respectively<sup>5</sup>. The figure shows BM prediction results for both models on an intact BM (Fig. 3 a, b) and a degraded BM (Fig. 3 c, d) from the test set. Figure 3 c shows that GraphSAGE+Mul erroneously classifies non-crossing edges close to the BM as crossing-edges, as shown by the many red lines. We term this tendency a “halo effect”. Notice that EAGNN+BC+MLP does not exhibit this halo effect as the false positives in Fig. 3 d are primarily in the region of broken BM.

## 6 Limitations

There exist several limitations to this study. One is the limitation of our dataset and prediction task: BM identification. Further cell-graph datasets and prediction tasks are needed to establish the wider value of the edge aggregation, classification and labelling methods proposed here.

Another general limitation is the data imbalance problem inherent in many medical datasets, due to low disease frequency, small sample sizes and the effort of annotation. A specific example of this limitation is the class imbalance problem between BM crossing and non-crossing edges. Since the BM is a thin protein interface localised between the epithelial and connective tissue, non-crossing BM edges constitute a large majority of the edges in each cell graph, as shown in Table 1. Figure 3 shows the difference in F1-scores for BM crossing edges between a healthy tissue sample where the BM is intact and a degraded sample where the BM is broken. This specific imbalance could be addressed by using the focal loss function [19] to reduce class imbalance.

## 7 Conclusions

In this work, we have proposed a new GNN model EAGNN that aggregates both node and edge label information and is suitable for edge label prediction. We have presented a digital pathology case study of BM prediction showing that aggregation of both node and edge label information can take advantage of the topological information in cell-graphs. In our case study, EAGNN significantly outperformed the widely used GraphSAGE GNN model under several performance measures. Furthermore, we have introduced new edge label features including cell entropy gradient and neighbourhood overlap similarity and shown that these improve the accuracy of BM prediction. Future directions of research include improving EAGNN performance and data augmentation methods to reduce the data imbalance.

**Acknowledgements.** The authors acknowledge the clinical and technical assistance of Karin Garming-Lergert, Victor Tollemar and Daniella Nordman (Karolinska Institutet) for collection and preparation of biopsies, and annotations. The study was

<sup>5</sup> We used [Intelligraph](#) tool for cell annotation, graph construction and results evaluation.

financed by grants from Region Stockholm ALF Medicine, Styrggruppen KI/Region Stockholm for Research in Odontology and research funds from Karolinska Institutet and KTH.

**Ethics Declarations.** Ethical approval for the collection and experimental use of tissue samples from patients attending the Department of Maxillofacial Surgery, Karolinska University Hospital (provided written informed consent) or retrieved retrospectively from Stockholm’s Medicine Biobank (Sweden Biobank, access to archived material) has been granted by the Swedish Ethical Review Authority (Etikprövningsmyndigheten). Procedures have been performed according to relevant guidelines, under Registration Numbers 2013/39-31/4, 2014/1184-31/1 and 2019-01295.

## References

1. Ahmedt-Aristizabal, D., Armin, M.A., Denman, S., Fookes, C., Petersson, L.: A survey on graph-based deep learning for computational histopathology. *Comput. Med. Imaging Graph.: The Official Journal of the Computerized Medical Imaging Society* **95**, 102027 (2022)
2. Anand, D., Gadiya, S., Sethi, A.: Histograms: graphs in histopathology. In: *Medical Imaging 2020: Digital Pathology*, vol. 11320, p. 113200O. International Society for Optics and Photonics (2020)
3. Atwood, J., Towsley, D.: Diffusion-convolutional neural networks. In: *Advances in Neural Information Processing Systems*, vol. 29 (2016)
4. Aygüneş, B., Aksoy, S., Cinbiş, R.G., Kösemehmetoğlu, K., Önder, S., Üner, A.: Graph convolutional networks for region of interest classification in breast histopathology. In: *Medical Imaging 2020: Digital Pathology*, vol. 11320, p. 113200K. International Society for Optics and Photonics (2020)
5. Bilgin, C., Demir, C., Nagi, C., Yener, B.: Cell-graph mining for breast tissue modeling and classification. In: *2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 5311–5314. IEEE (2007)
6. Cao, L., Lu, Y., Li, C., Yang, W.: Automatic segmentation of pathological glomerular basement membrane in transmission electron microscopy images with random forest stacks. *Comput. Math. Methods Med.* **2019** (2019)
7. Clevert, D.A., Unterthiner, T., Hochreiter, S.: Fast and accurate deep network learning by exponential linear units (ELUs). *arXiv preprint [arXiv:1511.07289](https://arxiv.org/abs/1511.07289)* (2015)
8. Crichton, G., Guo, Y., Pyysalo, S., Korhonen, A.: Neural networks for link prediction in realistic biomedical graphs: a multi-dimensional evaluation of graph embedding-based approaches. *BMC Bioinform.* **19**(1), 1–11 (2018)
9. Demir, C., Gultekin, S.H., Yener, B.: Learning the topological properties of brain tumors. *IEEE/ACM Trans. Comput. Biol. Bioinf.* **2**(3), 262–270 (2005)
10. Gilmer, J., Schoenholz, S.S., Riley, P.F., Vinyals, O., Dahl, G.E.: Neural message passing for quantum chemistry. In: *International Conference on Machine Learning*, pp. 1263–1272. PMLR (2017)
11. Glorot, X., Bordes, A., Bengio, Y.: Deep sparse rectifier neural networks. In: *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 315–323. JMLR Workshop and Conference Proceedings (2011)

12. Gong, L., Cheng, Q.: Exploiting edge features for graph neural networks. In: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Los Alamitos, CA, USA, pp. 9203–9211. IEEE Computer Society (2019)
13. Gunduz, C., Yener, B., Gultekin, S.H.: The cell graphs of cancer. *Bioinformatics* **20**(Suppl. 1), i145–i151 (2004)
14. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. In: *Advances in Neural Information Processing Systems*, vol. 30 (2017)
15. Hamilton, W.L.: Graph representation learning. *Synth. Lect. Artif. Intell. Mach. Learn.* **14**(3), 1–159 (2020)
16. Levy, J., Haudenschild, C., Barwick, C., Christensen, B., Vaickus, L.: Topological feature extraction and visualization of whole slide images using graph neural networks. In: *BIOCOMPUTING 2021: Proceedings of the Pacific Symposium*, pp. 285–296. World Scientific (2020)
17. Nair, A., et al.: A graph neural network framework for mapping histological topology in oral mucosal tissue. *BMC Bioinform.* **23**(1), 1–21 (2022)
18. Niepert, M., Ahmed, M., Kutzkov, K.: Learning convolutional neural networks for graphs. In: *International Conference on Machine Learning*, pp. 2014–2023. PMLR (2016)
19. Pasupa, K., Vatathanavaro, S., Tungjitnob, S.: Convolutional neural networks based focal loss for class imbalance problem: a case study of canine red blood cells morphology classification. *J. Ambient Intell. Hum. Comput.* 1–17 (2020)
20. Pati, P., et al.: HACT-Net: a hierarchical cell-to-tissue graph neural network for histopathological image classification. In: Sudre, C.H., et al. (eds.) *UNSURE/GRAIL -2020*. LNCS, vol. 12443, pp. 208–219. Springer, Cham (2020). [https://doi.org/10.1007/978-3-030-60365-6\\_20](https://doi.org/10.1007/978-3-030-60365-6_20)
21. Studer, L., Wallau, J., Dawson, H., Zlobec, I., Fischer, A.: Classification of intestinal gland cell-graphs using graph neural networks. In: 2020 25th International Conference on Pattern Recognition (ICPR), pp. 3636–3643. IEEE (2021)
22. Sureka, M., Patil, A., Anand, D., Sethi, A.: Visualization for histopathology images using graph convolutional neural networks. In: 2020 IEEE 20th International Conference on Bioinformatics and Bioengineering (BIBE), pp. 331–335. IEEE (2020)
23. Tollema, V., et al.: Histopathological grading of oral mucosal chronic graft-versus-host disease: large cohort analysis. *Biol. Blood Marrow Transplant.* **26**(10), 1971–1979 (2020)
24. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks. In: *International Conference on Learning Representations* (2018)
25. Wang, D., Gu, C., Wu, K., Guan, X.: Adversarial neural networks for basal membrane segmentation of microinvasive cervix carcinoma in histopathology images. In: 2017 International Conference on Machine Learning and Cybernetics (ICMLC), vol. 2, pp. 385–389. IEEE (2017)
26. Wu, H.S., Dikman, S., Gil, J.: A semi-automatic algorithm for measurement of basement membrane thickness in kidneys in electron microscopy images. *Comput. Methods Programs Biomed.* **97**(3), 223–231 (2010)
27. Yener, B.: Cell-graphs: image-driven modeling of structure-function relationship. *Commun. ACM* **60**(1), 74–84 (2016)