POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Convolution on Triangular Motifs: A GNN Approach to Edge Classification in Cell-Graphs

Tesi di Laurea Magistrale in
Computer Science and Engineering

Author: **Andrea Camilloni**

Student ID: 10866376
Advisor: Prof. Daniele Loiacono
Co-advisors: Karl Meinke, Rachael Sugars
Academic Year: 2022-23

Abstract

Introduction: Identifying the structure and health of tissues is crucial in digital pathology, and the basement membrane (BM) serves as a key indicator for evaluating tissue architecture. Existing Graph Neural Networks (GNNs) approaches to edge classification in cell-graph models lack true convolutional characteristics and may fail to account for recurrent structures inherent in the graph, such as triangular motifs. **Aim:** This thesis aims to introduce and evaluate a novel framework for edge classification in cell-graphs, focusing on identifying basement membranes (BMs) in oral mucosa samples. The model targets improved classification accuracy by incorporating convolutional structures and recurrent triangular motifs in a Delaunay-generated graph. **Methods:** To address this challenge, an Edge Aggregated Graph Neural Network (EAGNN) was employed for node and edge feature aggregation, and a Triangular Motif Convolutional Neural Network (TM-CNN) was introduced for edge classification. The dataset for this study comprises manually annotated oral mucosa samples from both healthy and diseased patients, provided by the Karolinska Institutet. The graph models for these samples were generated using Delaunay triangulation. **Results:** When compared to existing literature baselines, the proposed model exhibited superior performance, achieving a 2% higher F1 score for the minor class and significant improvements in identifying degraded BMs in tissue samples. **Conclusion:** The findings validate that incorporating convolutional layers and considering recurrent triangular motifs in the graph improves edge classification. The model's ability to identify BMs, even in degraded tissue structures, holds significant implications for diagnostic accuracy in oral pathology.

Keywords: Digital Pathology, Graph Neural Network, Convolutional Neural Network, Triangular Motif, Cell-Graph, Basement Membrana, Oral Mucosa

Abstract in lingua italiana

Nel contesto avanzato della patologia computazionale, la segmentazione dei tessuti è stata identificata come una delle questioni più complesse e fondamentali, in particolare l'identificazione delle membrane basali (BM). Questa problematica è considerata di importanza cruciale per gli esperti di patologia orale, per l'analisi delle biopsie e per la determinazione del grado istopatologico delle malattie. Per affrontare questo complesso problema, un approccio innovativo è stato sviluppato, in cui Edge Aggregated Graph Neural Network (EAGNN) è stata utilizzata per l'aggregazione delle caratteristiche di nodi e bordi, e Triangular Motif CNN (TM-CNN) è stata introdotta per catturare strutture ricorrenti nel grafo e fornire una classificazione accurata dei bordi,

Il dataset per la valutazione del modello è stato fornito dall'Istituto Karolinska e include un ampio numero di campioni di mucosa orale, che sono stati raccolti sia da pazienti sani che da pazienti affetti da varie malattie orali. Un'accurata annotazione manuale di ogni campione è stata effettuata da esperti in patologia orale. Un modello a grafo cellulare è stato costruito su queste annotazioni utilizzando algoritmi avanzati di triangolazione, specificamente la triangolazione di Delaunay.

Un rigoroso processo di valutazione è stato effettuato, durante il quale le prestazioni del modello sono state paragonate con altri metodi esistenti nella letteratura scientifica. È stato dimostrato che il modello sviluppato supera in prestazioni i metodi preesistenti, con un miglioramento particolarmente significativo evidente nei campioni con strutture di tessuto complesse o degradate. Un punteggio F1 superiore del 2% nella classe minoritaria è stato ottenuto dal modello, che ha anche dimostrato notevole robustezza nel riconoscimento delle membrane basali. In aggiunta, un metodo per la validazione dei risultati generati dai modelli è stato introdotto. Questo metodo ha contribuito notevolmente alla comprensione dei risultati. La segmentazione del grafo cellulare è stata effettuata in base alle strutture triangolari che mostrano una presenza di membrane basali. Grazie al metodo proposto, i risultati dei modelli sono stati comparati visualmente, permettendo una migliore comprensione dell'efficacia dell'approccio. Il modello proposto ha mostrato una riduzione significativa nelle previsioni rumorose, particolarmente nei campioni che presentano una struttura di BM degradata.

Non solo l'efficacia del modello nel migliorare la capacità di classificazione dei bordi tra diverse strutture cellulari è stata evidenziata, ma nuove frontiere per la comprensione e l'interpretazione delle interazioni tra cellule e BM in modelli a grafico cellulare sono state aperte. Vaste implicazioni per il campo della biologia computazionale e della medicina personalizzata potrebbero essere sostenute da questi risultati.

Future direzioni prevedono l'estensione del modello a altre tipologie di tessuti e malattie, con l'intenzione di migliorare ulteriormente la precisione e l'efficacia della patologia computazionale. Inoltre, l'integrazione del modello propostocon altre tecniche di imaging medico potrebbe essere considerata per fornire un quadro clinico più completo e dettagliato, contribuendo così a una diagnosi più accurata e a trattamenti più efficaci.

Parole chiave: Patologia Computazionale, Graph Neural Network, Convolutional Neural Network, Triangular Motif, Grafo Cellulare, Membrana Basale, Mucosa Orale

Contents

Abstract **i**

Abstract in lingua italiana **iii**

Contents **v**

1 Introduction **1**

2 State of the Art **5**

2.1 Medical Background 5

2.1.1 Oral Histology 5

2.1.2 Oral Tissue 6

2.1.3 Oral Mucosa 6

2.1.4 Digital Pathology (DP) 9

2.1.5 Oral cGVHD and Grading of Tissue Severity 10

2.2 Technical Background 10

2.2.1 Machine Learning (ML) 11

2.2.2 Artificial Neural Networks (ANNs) 13

2.2.3 Graph Theory 20

2.2.4 Graph Neural Networks (GNNs) 23

2.3 Related Work 28

2.3.1 Deep Learning in DP 29

2.3.2 BM Identification 31

2.3.3 Edge Classification Task 32

3 Methodology **35**

3.1 Graph definitions 35

3.2 KI dataset 37

3.3 Cell-graph model 38

3.4 GNN Framework . . . 40
3.4.1 Node Embedding Layers . . . 41
3.4.2 Edge Classifier: TM-CNN . . . 42
3.5 Model Optimization and Backpropagation . . . 45
3.5.1 Loss Function and Optimization . . . 45
3.5.2 Backpropagation . . . 45
3.6 Evaluation Metrics . . . 46
3.6.1 Precision . . . 46
3.6.2 Recall . . . 47
3.6.3 F1 score . . . 47
3.6.4 ROC-AUC Curve . . . 47
3.6.5 Accuracy . . . 47

4 Experiments and Results **49**
4.1 Experimental Setup . . . 49
4.2 Model Evaluation . . . 50
4.2.1 Validation Results . . . 51
4.2.2 Testing Results . . . 52
4.3 Qualitative Results . . . 53
4.3.1 Graph Segmentation . . . 54

5 Discussion **61**
5.1 Interpretations . . . 61
5.2 Implications . . . 62
5.3 Limitations . . . 63
5.4 Future Works . . . 64

6 Conclusions **67**

Bibliography **69**

A Appendix A **79**
A.1 Model Selection . . . 79
A.2 Ablation study . . . 80

B Appendix B **83**

B.1 Graph Segmentation 83

List of Figures **89**

List of Tables **91**

Acknowledgements **93**

1 | Introduction

As the digital revolution continues to impact healthcare, the field of digital pathology stands at an intriguing crossroads. Despite remarkable advancements in imaging and computational methods, the sector grapples with persistent challenges, particularly in automating and accurately identifying histological features crucial for medical diagnoses and treatment planning. One case in point is the identification of features like the basement membrane (BM) in complications arising from chronic graft-versus-host disease (cGVHD) after hematopoietic stem cell transplantation (HSCT). Traditional methods that rely on manual annotation of these features in whole-slide images (WSIs) are not only time-consuming but also prone to observer variability [68]. Thus, there is an urgent need for techniques that enhance efficiency and reduce errors in histological examinations.

In recent years, machine learning methods, particularly Graph Neural Networks (GNNs), have emerged as innovative tools to tackle this challenge. They have been utilized to automate and refine the accuracy of these histological analyses. Notably, cell-graph models that can capture intricate cellular interactions and high-level histological relationships in WSIs have been proposed [3, 27, 48]. While GNNs demonstrate significant promise in handling these cell-graph models to identify and classify complex histological features such as the BM [27, 48], they may fall short in capturing complex, higher-level structural features within graphs. This is an area where Convolutional Neural Networks (CNNs) excel, given their inherent ability to identify intricate patterns and structures. While GNNs focus on message-passing mechanisms, they may not capture these complex structural patterns within the graphs as effectively. Therefore, the integration of CNN-inspired concepts into GNNs could provide an innovative pathway to enhancement, leading to the capture of richer graph representations.

This work aims to address this limitation by incorporating a pattern recognition component into the GNN framework. This enhancement not only improves the accuracy and robustness of BM predictions, but also provides a valuable tool for pathologists. It automates the identification of BM, potentially assisting in the histological grading process and facilitating more efficient and precise diagnostics in oral pathology. This study was

carried out in collaboration with the KTH Royal Institute of Technology (KTH) and Karolinska Institute (KI).

Contributions of this Work Building upon the framework established by [27], this work maintains the core structure while introducing key enhancements. New features, extracted using CNNs pre-trained on existing datasets, aiming to enrich the representational capabilities of the cell-graph models, are introduced.

Given that conventional edge classification or link prediction approaches typically focus on the two nodes encompassing an edge, these models often miss out on a more comprehensive context.

To address this limitation, the main contribution of this work is the development of a new classifier that combines convolutional layers for pattern recognition with triangle based recurrent structures for edge classification. This classifier uses recurrent triangle based motifs associated with each edge in the cell-graph, taking a context-aware approach. The consideration of broader context provided by graph motifs has resulted in fewer false positives, effectively providing cleaner, more accurate results, and significantly enhancing the accuracy of BM identification.

Differing from earlier approaches [27, 48], the error was backpropagated through the entire structure (including the GNN and the classifier), resulting in a comprehensive learning mechanism. This mechanism marked a departure from the methodology in [27, 48], where the error was propagated only within the GNN layers. A notable enhancement in the performance of the GNN, especially for BM identification tasks, was achieved through this learning mechanism.

Furthermore, a graph segmentation technique is developed, offering a refined means of evaluating the GNN predictions on the cell-graph models, facilitating clearer interpretation and insights into model performance.

In conclusion, the proposed approach underscores the importance of a more context-aware, structure-oriented edge classification. This significantly enhances the performance of GNNs in Digital Pathology (DP) tasks, particularly in BM identification, marking a substantial advancement in this field.

Thesis Outline The rest of this thesis is organized as follows:

- **State of the Art:** In this chapter, the existing knowledge and developments in fields relevant to the research are presented. The medical background establishes foundational knowledge on oral histology, tissue, mucosa, and the technological

aspects of digital pathology. The technical background provides insight into machine learning, artificial neural networks, graph theory, and graph neural networks. The chapter concludes with a review of significant works related to deep learning in digital pathology, BM identification, and edge classification.

- **Methodology:** This chapter details the methods and approach taken in the research. The foundations of the graph definitions, datasets, and cell-graph modelling are laid out. Further, details of the GNN framework, model optimization techniques, and the evaluation metrics used are discussed in depth.
- **Experiments and Results:** Here, the practical implementation of the research is presented. The setup of the experiments, the results obtained, and a qualitative assessment are presented, focussing on graph segmentation.
- **Discussion:** In this chapter, reflections on the findings are presented. We interpret the results, discuss their broader implications, identify the limitations of the study, and propose potential avenues for future research.
- **Conclusions:** The main research takeaways are summarized, reaffirming the significance of the findings and their contribution to the field.

2 | State of the Art

This State of the Art section will provide a comprehensive overview of the existing research in the field of Machine Learning in DP and how it has been applied to the diagnosis and management of oral diseases. The section will be organized into three parts: Medical Background, Technical Background, and Related Works.

2.1. Medical Background

In this section, the medical background necessary for understanding the context of oral tissue analysis and the importance of image analysis is provided. It is essential to understand the basic structure and functions of oral tissues, particularly the basement membrane (BM), to appreciate the significance of oral tissue analysis in disease diagnosis and management. Equally important is the comprehension of oral diseases and their impact on these tissues

2.1.1. Oral Histology

Histology is the study of the microscopic structure of tissues and organs, and how these structures relate to their functions [52]. In the case of oral histology, this involves examining the cellular and extracellular components of the oral tissues, and understanding how they contribute to the various physiological and pathological processes that occur in the mouth [49].

There are many different techniques and tools that are used in oral histology, including microscopy, staining techniques, and imaging technologies. By using these tools, oral pathologists are able to gain a deeper understanding of the structure and function of the oral tissues, and to identify changes or abnormalities that may be indicative of disease or other health conditions [49].

A specific instance where oral tissue analysis proves to be crucial is in the case of Hematopoietic Stem Cell Transplantation (HSCT) patients who may develop oral chronic Graft-versus-Host Disease (cGVHD). cGVHD is a common complication post-transplantation,

often manifesting as a variety of symptoms such as mouth sores, dry mouth, and changes in taste. Histological examination of oral tissues in these patients can help in the early detection and diagnosis of oral cGVHD, thereby facilitating timely intervention and management [68].

2.1.2. Oral Tissue

Collecting an oral tissue involves a procedure known as a biopsy, which is the removal of a small sample of tissue from the oral cavity.

Once the samples are collected, they undergo a series of steps for preparation. The first step is fixation, which allows to preserve the sample structure. To prevent breakdown of DNA, RNA, proteins and tissue structure specimens are immersed in fixative [58].

The next step involves infiltration of the specimen with paraffin, which is an embedding medium that facilitates slicing of the tissue into thin sections, typically ranging from 5 to 15 μm, using a microtome [58]. Prior to downstream analyses, the paraffin is removed by treating the specimen with xylene.

The third step involves staining the colorless paraffin sections, using haematoxylin and eosin (H&E) to enable light microscopic examination. Firstly, the tissue is immersed in water with haematoxylin, which imparts a general deep purple or blue color to the specimen. Structures with high affinity for the dye, such as nuclear DNA and cytoplasmic RNA, are more heavily stained. Subsequently, the slide is stained with eosin, which provides a pink dye to proteins non-specifically [58].

The last step involves mounting the slides with pinene or acrylic resins as adhesives and covering them with a coverslip to create a permanent preparation suitable for analysis. This preparation can be viewed using a microscope or scanned using advanced equipment such as whole slide scanners to produce high-resolution digital images known as Whole Slide Images (WSIs), which can be viewed on a computer screen [58]. An example of H&E stained oral tissues is illustrated in figure 2.1.

2.1.3. Oral Mucosa

The oral mucosa is the lining of the oral cavity [49], which covers the lips, cheeks, tongue, hard and soft palates, and floor of the mouth. The oral mucosa has a number of purposes, such as protecting the oral cavity's deeper tissues, providing sensation, acting as the location of glandular activity, and acting as a conduit for secretion[50]. Anatomically, the oral mucosa is between the skin and the gastrointestinal mucosa and exhibits some traits

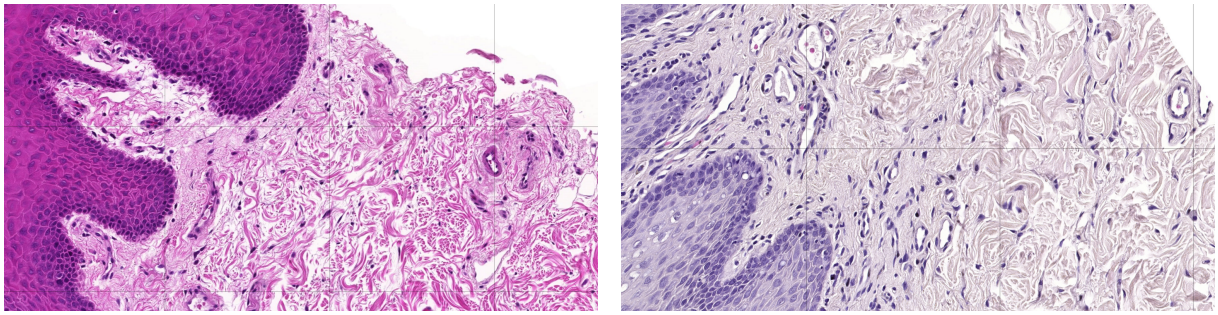


Figure 2.1: Two examples of H&E staining of oral mucosa samples

of both.

An underlying layer of connective tissue known as the lamina propria and a stratified squamous epithelium known as the oral epithelium make up the two major tissue components of the oral mucosa [49, 50].

Just like any other tissue, the oral mucosa can be in a state of health or disease. In a healthy state, it performs its functions effectively, maintaining the overall oral health. However, in the presence of certain conditions such as infections, autoimmune diseases, or malignancies, the oral mucosa can become inflamed. This inflammation can alter its structure and function, leading to symptoms like pain, redness, swelling, and sometimes even ulceration [68].

In figure 2.2, examples of oral mucosa in both states - healthy and inflamed - are depicted. This comparison serves to illustrate the changes that occur in the oral mucosa during disease processes.

Oral Epithelium

The oral epithelium is a stratified squamous epithelium that is involved in protecting the oral cavity against physical, chemical, and microbial challenges. It also plays a role in sensation and secretion. The oral epithelium undergoes continuous turnover and regeneration, which is essential for maintaining the integrity of the oral mucosa [49].

Basement Membrane (BM)

The basement membrane (BM) is a thin sheet-like structure that separates the epithelial layer from the underlying connective tissue (lamina propria). It is crucial in supporting the epithelium, facilitating cell migration during wound healing, and acting as a barrier. The BM is mainly composed of collagen, laminins, and other proteins.

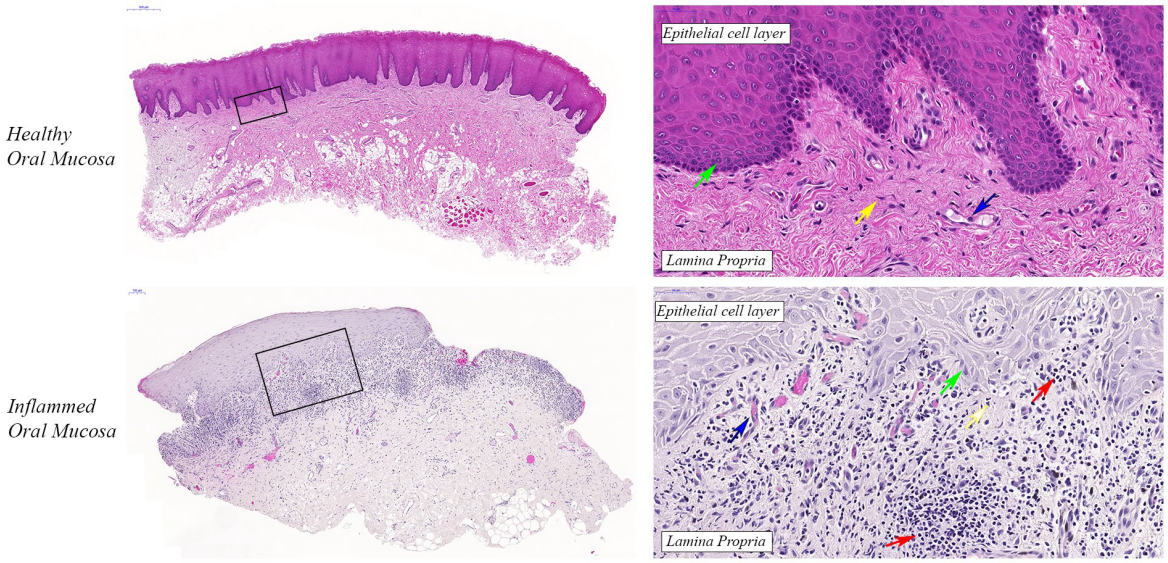


Figure 2.2: Whole slide images (WSIs) of a healthy oral mucosa (top) and inflammed oral mucosa (bottom). The boxes show the magnified areas. Arrows point to key cell types, epithelial tissue (green), fibroblasts (yellow), endothelial cells (blue) and inflammation (red). The finger structure of the lamina propria can be seen as well as the basal layer of the epithelial. The very densely packed cells of the basal layer can be seen in the magnified images and how they diverge from each other when approaching the top.

Disruptions or breakages in the BM can indicate pathological conditions, allowing cells such as cancer cells or inflammatory cells to infiltrate deeper tissues. This is commonly associated with the progression of diseases like oral cancer [68]. An example of a disrupted BM can be seen in the bottom figure of figure 2.2, where the continuity of the BM is compromised.

In the context of oral chronic Graft-versus-Host Disease (cGVHD), the BM plays a significant role. In this condition, the immune cells of the donor graft attack the recipient's tissues, including the oral mucosa. The BM, in this case, can become a site of immune attack, leading to its disruption and subsequent tissue damage. Understanding the changes in the BM in oral cGVHD can provide valuable insights into the disease's pathogenesis and help in the development of targeted therapeutic strategies.

Lamina Propria

The lamina propria is the connective tissue layer found under the epithelium. It provides the oral mucosa with elasticity and resilience. The lamina propria also contains blood vessels, nerve endings, and immune cells, which are essential for nourishment, sensation,

and defense against pathogens.

2.1.4. Digital Pathology (DP)

Pathology is the study of disease, including the causes, nature, and effects of disease on the body [18]. In medicine, pathology plays a critical role in the diagnosis and treatment of various diseases. Traditionally, pathologists analyze tissue samples under a microscope to identify abnormalities and make a diagnosis.

Digital Pathology (DP), also known as virtual pathology or telepathology, is a relatively new field in pathology that uses digital technology to analyze and interpret pathology images [34]. In DP, glass slides containing tissue samples are scanned to create WSIs that can be viewed on a computer or mobile device. These images can then be analyzed and interpreted by pathologists using specialized software.

DP is important for several reasons. First, it allows pathologists to view and analyze pathology images from anywhere in the world, which is particularly useful in areas where there is a shortage of pathologists [34]. Second, digital images can be easily shared with other medical professionals, allowing for more collaborative and accurate diagnoses [34]. Finally, DP has the potential to improve diagnostic accuracy and efficiency, leading to better patient outcomes [34].

Whole Slide Image (WSI)

WSI technology has revolutionized DP and opened up new avenues for more accurate and efficient diagnostic analysis. WSI offers numerous advantages over traditional microscopy, including the ability to capture high-resolution images of entire specimens, providing the pathologist with a more comprehensive view of the sample. With WSI, a pathologist can zoom in and out, navigate and annotate images, and share them with colleagues from anywhere, regardless of geographic location[34].

The advent of WSI has also paved the way for advanced image analysis techniques. These techniques, often powered by machine learning and artificial intelligence, can process and analyze the high-resolution images generated by WSI to extract meaningful information.

WSI has been shown to be especially beneficial when combined with computer-aided detection (CAD) tools, which can help detect subtle changes in tissue morphology that may not be apparent to the naked eye[25]. CAD systems can be trained using machine learning algorithms to accurately identify abnormal tissue patterns and classify them accordingly. This can greatly improve the accuracy and speed of diagnoses, as well as

reducing the risk of error associated with manual analysis.

In essence, the integration of WSI with advanced image analysis techniques is transforming the field of pathology, enabling more precise, efficient, and reproducible diagnoses [3, 16].

2.1.5. Oral cGVHD and Grading of Tissue Severity

Chronic Graft versus Host Disease (cGVHD) is a complication that may occur after Hematopoietic Stem Cell Transplantation (HSCT), a procedure that involves the transplantation of stem cells or bone marrow. In cGVHD, the donated cells perceive the recipient's body as foreign and initiate an immune response, attacking the recipient's tissues. The oral cavity is frequently affected in cGVHD, and oral manifestations can include changes in the oral mucosa, salivary glands, and other structures.

Grading systems have been developed to assess the severity of oral cGVHD based on histological criteria such as lymphocyte infiltration, inflammation, and BM breakages. These grading systems can help in making an objective assessment of the tissue and guiding the management and prognosis of patients undergoing HSCT. Table 2.1 presents a comprehensive grading system detailing the histological criteria used to assess the severity of oral mucosal cGVHD. By understanding the changes in the BM and other oral tissues as depicted in this grading system, clinicians can better predict the course of the disease and tailor treatment strategies accordingly [68].

2.2. Technical Background

This section serves as an introduction to the wide range of technical concepts that form the base of the present research. These diverse set of concepts and methodologies are fundamental to fully grasp the depth and breadth of the study at hand.

The discussion begins with the field of Machine Learning (ML), emphasizing its role in classification problems. This provides a transition into the realm of Artificial Neural Networks (ANNs), from their origins as perceptrons to their evolution into more sophisticated forms like Convolutional Neural Networks (CNNs), a cornerstone of modern deep learning architectures.

The discourse then ventures into Graph Theory and its embodiment in Graph Neural Networks (GNNs). This includes an examination of the core principles of GNNs and how they extend the foundational concepts of ANNs to handle structured data in graph form. The specific mechanics of message passing in GNNs, along with their types and uses, are also explored.

	Grade				
Features	0	1	2	3	4
1. Inflammation infiltrate	Small numbers of scattered cells, papilla focused	Sparsely clustered cells	Tightly clustered cells	Band-like infiltrate	Extensive band-like infiltrate
Score:	**0**	**1**	**2**	**3**	**4**
2. Intra-epithelial infiltration of lymphocytes	None / occasional	Sporadic	Focal	Widespread	
Score:	**0**	**1**	**2**	**3**	
3. Liquefaction degeneration	None	Sporadic	Widespread	Confluent	
Score:	**0**	**1**	**2**	**3**	
4. Apoptosis	None / occasional	Sporadic	Widespread		
Score:	**0**	**1**	**2**		
5. Basal membrane alteration	Thin intact	Increased thickness			Thinning / loss with detachment, and / or pseudo rete ridges
Score:	**0**	**1**			**4**
6. Flattening / atrophy	Normal rete ridges	Some flattening of rete ridges across biopsy (<25%)	Flattening of rete ridges across biopsy (25-75%)	Flat atrophic oral epithelia across biopsy (75-100%)	
Score:	**0**	**1**	**2**	**3**	
					Total score: 19

Table 2.1: Histological grading criteria for defining features of oral mucosal cGVHD [68]

Building on this, the spotlight shifts to Cell Graphs in DP, examining their construction and utility. This sets the foundation for a discussion on the crucial role and techniques of edge classification in graph analysis.

In summary, this section seeks to furnish the reader with the technical knowledge required to understand the research methods, significance of the outcomes, and the overall contribution to the field. It also lays the groundwork for the upcoming "Related Work" section, which situates this research within the wider context of existing studies in this domain.

2.2.1. Machine Learning (ML)

Machine Learning (ML) is a field of artificial intelligence that focuses on the design of systems that can learn from and make decisions based on data. ML systems are designed to learn from data, iteratively improve their performance, and adapt to new situations, reducing the need for explicit programming for every possible scenario. These systems employ algorithms that learn patterns within data and make predictions or decisions based on these recognized patterns [46].

Fundamental Concepts in ML

ML revolves around several core concepts. One fundamental concept is the idea of a *model*, a mathematical or computational structure that represents the patterns identified in the data. The learning algorithm generates this model based on the input data or *training data*. The quality of a model's predictions or decisions is measured by a *loss function*, which the algorithm strives to minimize. A *feature* refers to an individual measurable property or characteristic of the phenomenon being observed. Features constitute the dimensions of the model's input space, allowing it to learn from various aspects of the data [46].

Supervised and Unsupervised Learning The two primary categories of ML are supervised and unsupervised learning. In *supervised learning*, models are trained on a labeled dataset, i.e., each instance in the training data consists of an input vector and the corresponding target output. The goal is to learn a mapping from inputs to outputs and to be able to make accurate predictions for new, unseen data. Examples of supervised learning tasks include regression (predicting a continuous output) and classification (predicting discrete, categorical output) [8].

Unsupervised learning, on the other hand, involves learning from unlabeled data. The learning algorithm aims to discover the inherent structure or patterns in the input data. Examples of unsupervised learning tasks include clustering (grouping similar instances together), dimensionality reduction (simplifying the input data without losing too much information), and anomaly detection (identifying unusual instances) [8].

Overview of Classification Problems in ML Classification, a subset of supervised learning, is a task that involves predicting the class or category of an instance given its features. For example, in the context of email, the task might be to classify emails as "spam" or "not spam" based on their content. In the biomedical field, classification tasks often involve predicting the presence or absence of a disease based on patient data [8].

In binary classification, there are two possible classes, while in multiclass classification, there are more than two possible classes. Classification models are trained using a set of labeled instances, and their performance is evaluated based on their ability to correctly predict the classes of new, unseen instances. Some of the common metrics used for evaluating classification models include accuracy, precision, recall, and the F1 score [8].

Overview of Regression Problems in ML Regression, another significant area of supervised learning, involves predicting a continuous output variable given a set of input

features. For instance, predicting the price of a house based on various attributes like its size, location, number of rooms, and age is a classic example of a regression task. In a biomedical context, regression tasks often include predicting a patient's blood pressure, cholesterol levels, or any other quantitative biomarker based on personal and clinical data [8].

Regression models are trained using a dataset where the true output values are known, and their performance is typically evaluated based on the difference between the predicted and actual values. Common metrics for evaluating regression models include the mean absolute error (MAE), root mean square error (RMSE), and coefficient of determination (R^2) [8].

2.2.2. Artificial Neural Networks (ANNs)

Artificial Neural Networks (ANNs) are computational models inspired by the human brain's interconnected network of neurons. They are designed to learn patterns in data through an iterative process, enabling them to perform tasks such as classification, regression, and pattern recognition [22].

Basics of ANNs: Perceptron, Multi-layer Perceptron

The simplest form of an ANN is a *Perceptron*, proposed by Frank Rosenblatt in 1958 [57]. A perceptron takes a vector of input features and calculates a weighted sum, applying a step function to output a binary result. Figure 2.3 illustrates a basic perceptron model.

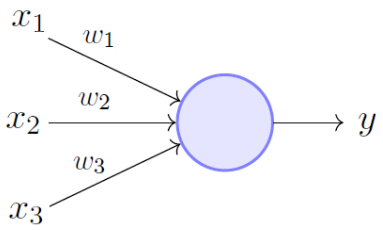


Figure 2.3: A basic Perceptron model.

A *Multi-layer Perceptron* (MLP) is a more complex ANN, comprised of multiple layers of perceptrons, or neurons. MLPs contain an input layer, one or more hidden layers, and an output layer. Each layer is fully connected to the next, with each connection having an associated weight [22]. MLPs can model non-linear relationships, making them more flexible and powerful than a single-layer perceptron. Figure 2.4 depicts a MLP with one hidden layer.

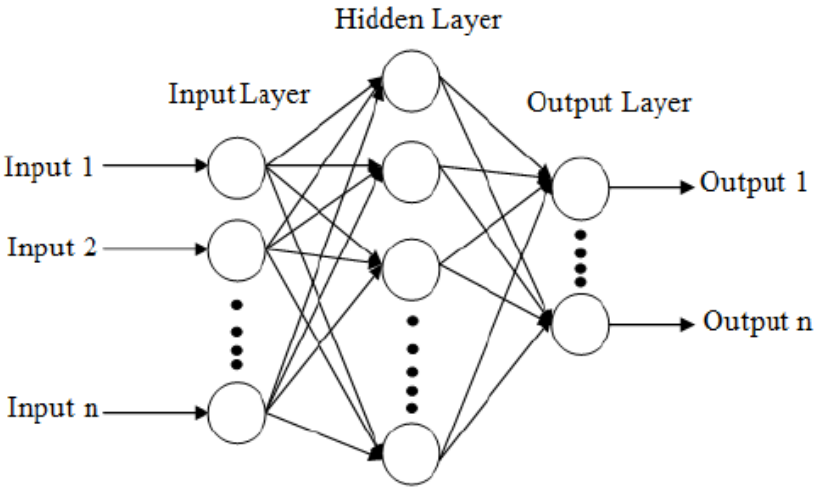


Figure 2.4: A Multi-layer Perceptron (MLP) with one hidden layer.

The non-linear relationship between layers in MLPs is introduced through *Activation Functions*. They perform a non-linear transformation of the input data, enabling the model to learn complex patterns. Some commonly used activation functions include the Sigmoid, Tanh, and ReLU [2] (Rectified Linear Unit) functions.

Loss Functions quantify the difference between the network's predictions and the actual values. They are used during the training phase to update the network's weights, with the goal of minimizing this difference. Commonly used loss functions include the Mean Squared Error (MSE) for regression tasks and Cross-Entropy Loss for classification tasks.

These elements together form the foundational building blocks of ANNs.

Activation Functions

Activation functions introduce non-linearity into the network, enabling it to learn complex patterns and make more nuanced predictions. Without an activation function, a neural network would only be capable of learning linear relationships, regardless of its depth. Some commonly used activation functions include:

- **Sigmoid:** The sigmoid function maps any input to a value between 0 and 1, which can be interpreted as a probability. Its mathematical form is:

$$f(x) = \frac{1}{1 + e^{-x}} \tag{2.1}$$

- **Tanh:** The hyperbolic tangent function, or tanh, also squashes its input but in the

range from -1 to 1. Its mathematical form is:

$$f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{2.2}$$

- **ReLU (Rectified Linear Unit):** The ReLU [2] function takes the maximum of 0 and the input value, effectively removing negative values during the learning process. ReLU and its variants (like Leaky ReLU, Parametric ReLU, and Exponential Linear Units) are the most commonly used activation functions in modern deep learning models. Its mathematical form is:

$$f(x) = \max(0, x) \tag{2.3}$$

Loss Functions

In the context of ML, a loss function L is used to quantify how far the predictions of a model, parameterized by θ, deviate from the true values given input data X and corresponding labels y. This can be represented as:

$$L(\theta, X, y) \tag{2.4}$$

The specific form of L depends on the task at hand. Some common choices include:

- **Mean Squared Error (MSE):** Used predominantly for regression tasks, the MSE loss function minimizes the difference between the predicted and actual continuous values, represented as:

$$L(y, \hat{y}) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{2.5}$$

- **Cross-Entropy Loss:** Typically used for binary or multi-class classification tasks, the Cross-Entropy loss function is formulated as:

$$L(y, \hat{y}) = -\frac{1}{n}\sum_{i=1}^{n} y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i) \tag{2.6}$$

Here, y represents the true label, $\hat{y}$ the predicted output, and n the total number of data points.

Backpropagation and Training of ANNs

Training an ANN involves adjusting the weights associated with each connection in the network to minimize the difference between the network's predicted output and the actual output, quantified by the loss function [22]. The mechanism used for this purpose is the Backpropagation algorithm, which is based on the concept of gradient descent.

Gradient descent adjusts the weights in the direction that reduces the loss function most rapidly, i.e., the negative of the gradient of the loss function with respect to the weights [59]. The Backpropagation algorithm, thus, comprises two main steps:

Forward Pass: The input data is propagated through the network layer by layer, and the output of each layer is computed using the inputs, weights, and activation function. The network's output is then compared with the actual output to compute the loss.

Backward Pass: The gradient of the loss function with respect to the weights is computed using the chain rule of calculus, a process known as backpropagation of errors. The weights are then updated by a small step in the direction of the negative gradient to decrease the loss [59].

This process is repeated over multiple epochs or full passes through the training dataset until the model's performance on a separate validation set stops improving, indicating convergence [22]. Through iterative minimization of the loss function via backpropagation and gradient descent, the network learns the optimal weights for accurate predictions on unseen data.

Model Regularization and Overfitting

Overfitting is a common problem in ML where a model performs well on the training data but poorly on unseen data (i.e., validation or test data). This typically occurs when the model learns the noise or outliers in the training data, making it overly complex and less generalizable [22]. Several techniques have been developed to prevent overfitting and improve model generalization, including regularization, dropout, batch normalization, early stopping, etc.

Regularization Techniques Regularization techniques add a penalty term to the loss function to discourage over-complex models and thus mitigate overfitting. The two most commonly used types of regularization are L1 and L2 regularization.

L1 regularization, or Lasso regression, adds the absolute value of the weights to the loss function. This can result in some weights being zero, effectively performing feature

selection [67]. Mathematically, the L1 regularized loss function can be expressed as:

$$L'(\theta, X, y) = L(\theta, X, y) + \lambda \sum |\theta| \tag{2.7}$$

L2 regularization, or Ridge regression, adds the squared value of the weights to the loss function. This generally results in smaller weights but does not drive them to zero [29]. The L2 regularized loss function can be represented as:

$$L'(\theta, X, y) = L(\theta, X, y) + \lambda \sum \theta^2 \tag{2.8}$$

These regularization techniques can make the model more robust to noise in the training data, improving generalization to unseen data.

Dropout Dropout is a specific regularization technique used in neural networks. During training, certain neurons are randomly "dropped out" or deactivated with a certain probability. This technique can prevent complex co-adaptations of the neurons, forcing them to learn more robust features that are useful in conjunction with many different random subsets of other neurons [64].

Dropout essentially creates a "thinned" version of the network, and the predictions for the final model are made by averaging the predictions of these thinned networks. This encourages the network to develop a redundant representation, making it more robust and reducing overfitting.

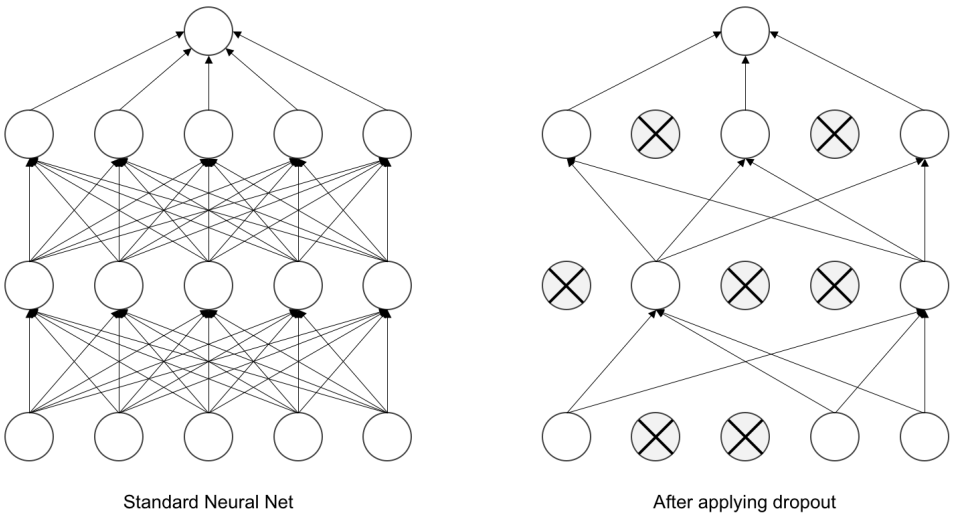


Figure 2.5: Illustration of a neural network with and without Dropout.

Figure 2.5 illustrates the concept of Dropout in a neural network. The left side represents a standard neural network with three hidden layers, while the right side represents the same network after applying Dropout. The crossed nodes are the "dropped out" neurons.

Batch Normalization Batch Normalization is another technique used to improve the training of ANNs. Introduced by Ioffe and Szegedy [30], this method addresses the problem of internal covariate shift, which is the change in the distribution of network activations due to the change in network parameters during training. This shift can slow down the training process and make it harder to achieve a stable network.

Batch Normalization addresses this issue by normalizing layer inputs for each mini-batch. This normalization helps to maintain a stable distribution of activations, thereby accelerating the training process. Additionally, Batch Normalization also has a regularizing effect similar to Dropout, as the normalization is based on the batch's statistical properties, adding some level of noise to the layer's inputs. This regularizing effect can reduce the need for Dropout or other regularization methods [30].

The batch normalization process involves two steps:

- Standardizing the inputs: The mean and variance for each input are computed across the mini-batch, and the inputs are then standardized.
- Scaling and shifting: The standardized inputs are then scaled and shifted using parameters that are learned during the training process.

The process of Batch Normalization can be represented as follows:

$$BN(x_i) = \gamma \left(\frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \right) + \beta \tag{2.9}$$

where x_i is the input, μ_B and σ_B^2 are the mean and variance of the mini-batch, ϵ is a small number to avoid division by zero, and γ and β are learned parameters for scaling and shifting.

By incorporating these techniques, it is possible to significantly reduce overfitting in ML models, leading to improved generalization and performance on unseen data.

Deep Learning and Convolutional Neural Networks (CNNs)

Deep Learning, a subset of ML, leverages ANNs with multiple layers, known as Deep Neural Networks (DNNs). These networks are capable of learning high-level features

from raw input data, making them particularly suited for tasks such as image and speech recognition, where the data has a complex hierarchical structure [40].

Convolutional Neural Networks (CNNs) are a specific type of DNN designed to process grid-like data, such as images. A CNN comprises convolutional layers, pooling layers, and fully connected layers. Convolutional layers apply a set of learnable filters to the input, pooling layers reduce the spatial dimensions, and fully connected layers perform classification. This architecture enables the CNN to automatically learn hierarchical feature representations from the raw image data [39]. Figure 2.6 shows a simplified architecture of a CNN.

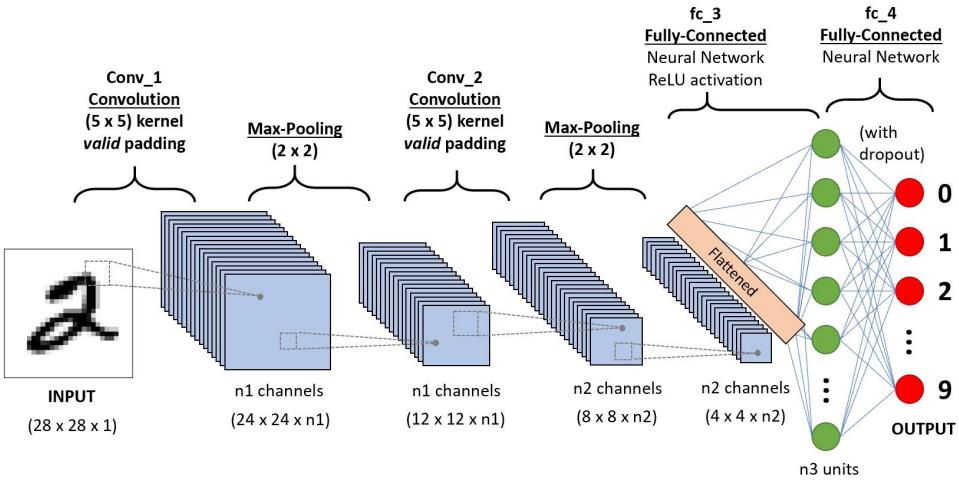


Figure 2.6: Simplified architecture of a Convolutional Neural Network (CNN).

Mathematical Framework of CNNs The operations performed in a CNN can be broken down into several key steps: convolution, activation, pooling, and classification/regression. These operations are represented mathematically as follows:

Convolution Operation The convolution operation, performed in the convolutional layer, applies a set of learnable filters (or Kernels) to the input. Each filter slides over the entire image and performs an element-wise multiplication followed by a summation (dot product) between the filter and the section of the image it currently covers. Mathematically, this operation is expressed as:

$$f_{i,j} = \sum_m \sum_n I_{i-m,j-n} K_{m,n} \tag{2.10}$$

where I is the input image, K is the filter, and $f_{i,j}$ is the feature map at location i, j. See Fig. 2.7.

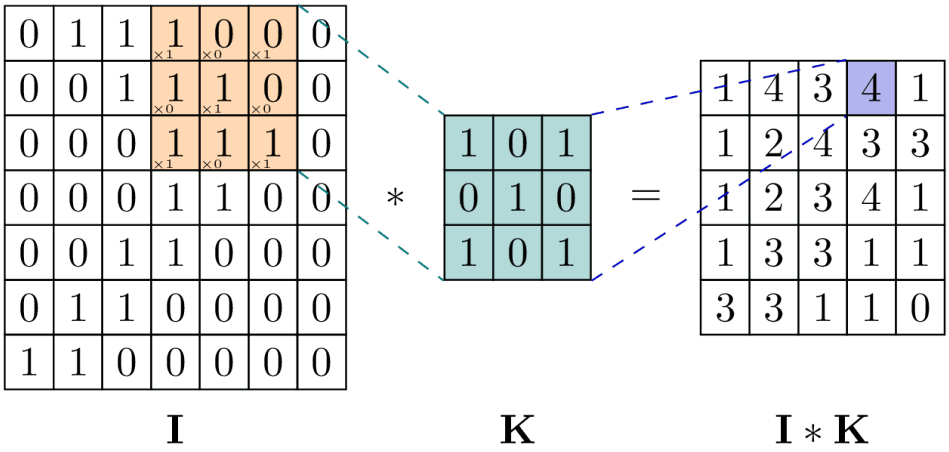


Figure 2.7: An illustration of the convolution operation. A 3x3 filter K slides over the input image I, and for each position, an element-wise multiplication followed by a summation (dot product) is performed to produce the feature map $I * K$.

Activation Function Post convolution, an activation function is applied to introduce non-linearity into the model. The most commonly used activation function in CNNs is ReLU [39].

Pooling Operation The pooling layer reduces the spatial dimensions (i.e., width and height) of the input. The most common type of pooling is max pooling, which takes the maximum value in the window (filter or kernel) that slides over the input. An example of pooling is shown in Figure 2.8.

Fully Connected Layer The final layers in a CNN are the fully connected layers, which perform classification/regression based on the features learned by the convolutional layers.

In the context of this study, deep learning, particularly CNNs, forms the foundational concepts that are extended to GNNs, paving the way for efficient edge classification in cell-graphs.

2.2.3. Graph Theory

Graph theory is a branch of mathematics that studies the properties of graphs. A graph, in this context, is a structure that consists of a set of vertices or nodes and a set of edges that connect pairs of vertices [43].

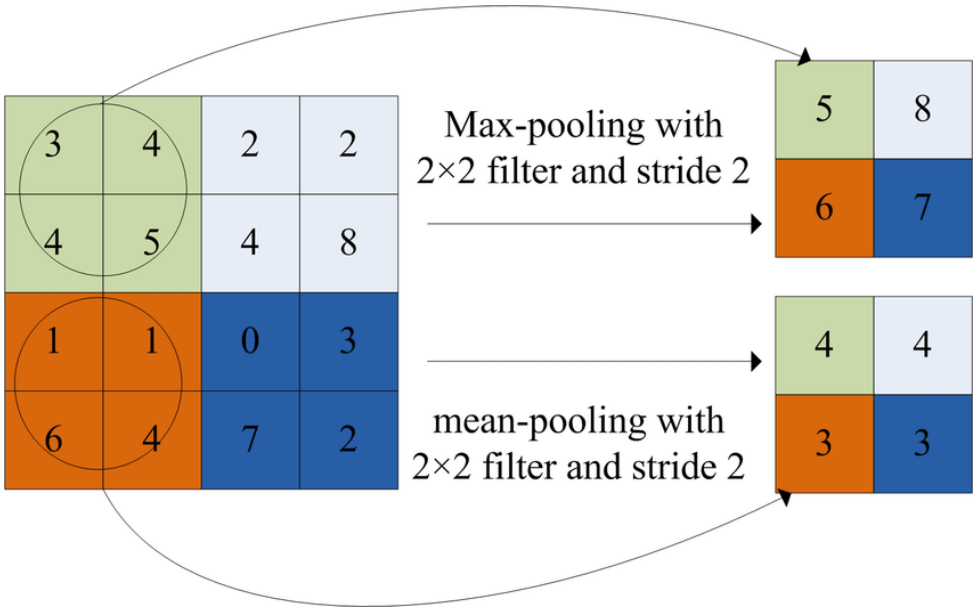
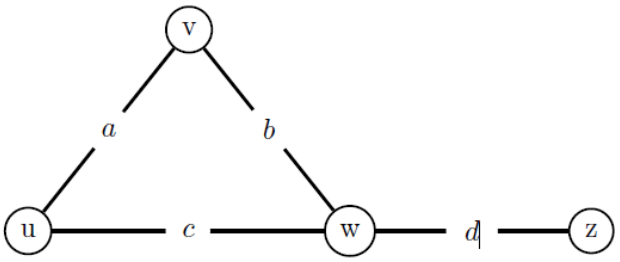


Figure 2.8: An illustration of the pooling operation with 2×2 filter and stride 2. (Top-Right) Result of Max-pooling. (Bottom-Right) Result of Mean-Pooling.

Mathematically, a graph G can be defined as $G = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is the set of vertices with $|\mathcal{V}| = n$ nodes and $\mathcal{E}$ is the set of edges connecting these nodes. An edge between two nodes $u, v \in \mathcal{V}$ is denoted as $(u, v) \in \mathcal{E}$.

One key concept in graph theory is the adjacency matrix, $A \in \mathbb{R}^{n \times n}$. For a graph with n vertices, A is an $n \times n$ matrix where $A[u, v] = 1$ if $\forall u, v \in \mathcal{V} : (u, v) \in \mathcal{E}$, otherwise 0 . For an undirected graph, the adjacency matrix is symmetric [43]. In fig. 2.9 an example of how an adjacency matrix is built is shown.

In case of a weighted graph the entries of the adjacency matrix can be arbitrary real values.



Adjacency Matrix

	u	v	w	z
u	0	1	1	0
v	1	0	1	0
w	1	1	0	1
z	0	0	1	0

Figure 2.9: *(Left)* Example of undirected graph with four nodes and four edges. *(Right)* Adjacency matrix of the graph.

Graphs can also have attributes and features at a node, edge or graph-level. Node-level

features can be represented by a feature matrix $X \in \mathbb{R}^{n \times d}$, while edge-level features by a tensor $E \in \mathbb{R}^{n \times n \times p}$, where m and p are respectively the dimension of a single node features vector and a single edge features vector.

ML Tasks on Graph Data

There are three main types of tasks that can be performed on graph data in the context of ML: node classification, link prediction or edge classification, and graph classification [43].

Node Classification Node classification is a key ML task that is applied to graph data. In this context, we are given a set of nodes, denoted as $\mathcal{V}$, and the goal is to predict the label y_u for each node $u \in \mathcal{V}$. This task differs from traditional supervised classification tasks where data points are assumed to be independent and identically distributed (i.i.d). In contrast, nodes in a graph are interconnected, forming a network of relationships that can be utilized to enhance the classification process [43].

In mathematical terms, the node classification task can be represented as: $y_u = f(x_u)$, where x_u denotes the feature vector of a node u and f is a function that can be learned from the data. The function f maps the features of each node to its corresponding label, thereby enabling the prediction of labels for unlabeled nodes.

Link Prediction and Edge Classification Link prediction is a task that aims to identify missing or potential connections (edges) in a graph. Given a set of nodes, represented as $\mathcal{V}$, and an incomplete set of edges, denoted as $\mathcal{E}$train, the objective is to predict the edges that are not present in $\mathcal{E}$train but exist in the complete edge set $\mathcal{E}$ [43]. This task can be viewed as both a supervised and unsupervised learning problem, depending on the availability of training data.

Mathematically, the link prediction task can be expressed as: $y_{uv} = f(x_u, x_v, e_{uv})$, where x_u and x_v are the feature vectors of nodes u and v, e_{uv} represents the edge between nodes u and v, and f is a function that can be learned to predict whether an edge y_{uv} exists between nodes u and v.

Edge classification is a related task, with the difference being that instead of predicting the existence of edges, it involves predicting the labels y_{uv} for each edge e_{uv} in $\mathcal{E}$. This task is useful in scenarios where edges in a graph can have different types or categories, and the goal is to classify these edges based on their characteristics.

Graph Classification, Regression, and Clustering These tasks involve classification, regression, or clustering over entire graphs. For classification or regression tasks, given a dataset of multiple different graphs, the goal is to make independent predictions specific to each graph. Graph clustering, on the other hand, involves learning an unsupervised measure of similarity between pairs of graphs.

In the context of graph classification or regression, the task can be mathematically expressed as: $y_G = f(\bigoplus_{u \in \mathcal{V}} x_u)$

2.2.4. Graph Neural Networks (GNNs)

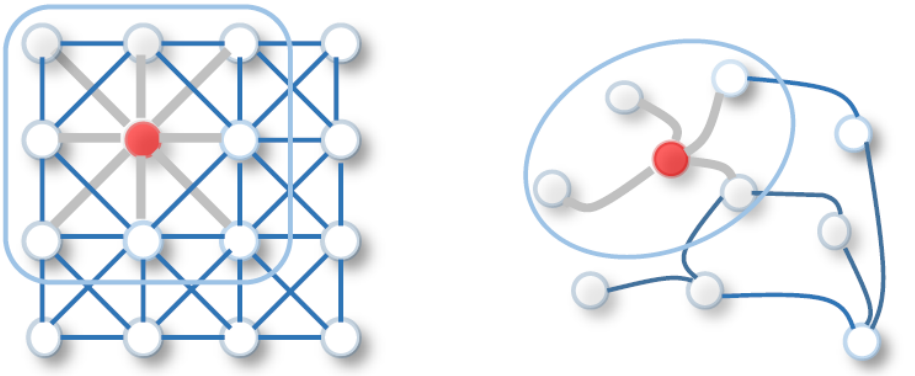
Introduced by Scarselli et al. in 2009, GNNs provide a way to apply deep learning techniques to graph-structured data [61]. GNNs are designed to accommodate certain characteristics of graph data that differentiate it from other forms of data [43]. One such property is that nodes in a graph don't have a specific order, making them permutation invariant. This characteristic requires us to use adjacency matrices that are independent of the node order [10]. Another factor is that graphs can differ in size and structure, unlike images that can be resized or cropped. GNNs address these characteristics by adopting a version of the convolution operation that can be applied to graph-structured data. They achieve this by implementing a form of message passing between nodes, updating information iteratively using simple non-linear neural network functions. This contrasts with the conventional convolution used in CNNs, as demonstrated in Figure 2.10.

At the start, the GNN layer accepts a graph $G = (\mathcal{V}, \mathcal{E})$ and a set of node features $X \in \mathbb{R}^{n \times d}$, where d symbolizes the embedding (features) dimension. The GNN aims to learn and adjust the node features X_v for each node $v \in \mathcal{V}$ by considering its own and neighboring features. This learning and updating process results in the final node embeddings $Z \in \mathbb{R}^{n \times d_k}$, where d_k is the dimension of the node embedding after k iteration.

At each iteration k, each node $u \in V$, represented by a hidden embedding $h_u^{(k)}$, gathers messages from nodes in its neighbourhood $N(u)$. This information is accumulated into a message $m_{N(u)}^{(k)}$ via the $AGGREGATE^{(k)}$ function, a permutation invariant operation that could be a sum, mean, or maximum:

$$m_{N(u)}^{(k)} = AGGREGATE^{(k)}(h_v^{(k)}, \forall v \in N(u)) \tag{2.11}$$

The collected message $m_{N(u)}^{(k)}$ is then merged with the earlier embedding $h_u^{(k-1)}$ of node u using the $UPDATE^{(k)}$ (denoted also as $COMBINE^{(k)}$) function, which results in a refreshed embedding $h_u^{(k+1)}$:



(a) 2D Convolution. Analogous to a graph, each pixel in an image is taken as a node where neighbors are determined by the filter size. The 2D convolution takes the weighted average of pixel values of the red node along with its neighbors. The neighbors of a node are ordered and have a fixed size.

(b) Graph Convolution. To get a hidden representation of the red node, one simple solution of the graph convolutional operation is to take the average value of the node features of the red node along with its neighbors. Different from image data, the neighbors of a node are unordered and variable in size.

Figure 2.10: Two-dimensional convolution vs graph convolution [75]

$$h_u^{(k+1)} = UPDATE^{(k)}(h_u^{(k)}, m_{N(u)}^{(k)}) \tag{2.12}$$

This procedure is executed for K iterations until the final layer outputs the node embedding $z_u \in \mathbb{R}^{d_k}$ for each node:

$$z_u = h_u^{(K)}, \forall u \in \mathcal{V} \tag{2.13}$$

The newly generated node embeddings serve as abstract representations learned through the accumulation of neighboring features within the nodes. These embeddings can be used to perform tasks, e.g., node classification and edge classification.

GNNs take various forms, each tailored to address specific challenges associated with processing graph-structured data. While all GNN architectures aggregate information from local neighborhoods to generate node embeddings, the mechanisms they employ to do so vary significantly.

Spatial-based GNNs: These models handle the graph data directly in its original form. The information from each node's neighborhood is aggregated and processed, considering the spatial relationships among the neighboring nodes [43].

Spectral-based GNNs: These models deal with the graph data in the spectral domain by using graph Fourier transform. They leverage the spectral theory of graphs, specifically the graph Laplacian and its eigenvalues and eigenvectors [43].

Spatial-based GNNs

The focus of this work is placed exclusively on the spatial-based formulation of GNNs. The framework for this formulation is presented in the subsequent paragraphs.

Mathematical Formulation In a typical spatial-based GNN, the architecture is layered, and each layer k computes a low-dimensional (d_k-dimensional) representation $h_u^k \in \mathbb{R}^{d_k}$ of the graph structure around node u. The computation at each layer k consists of two stages.

Firstly, an AGGREGATE^k operation produces an integrated representation $h_{N(u)}^k$ of all immediate neighbors $v \in N(u)$ of u using the representations h_v^{k-1} from the previous layer $k-1$:

$$h_{N(u)}^k = \text{AGGREGATE}^k(h_v^{k-1}, \forall v \in N(u)) \tag{2.14}$$

Secondly, a COMBINE^k operation updates the representation h_u^k of u by combining its previous representation h_u^{k-1} from layer $k-1$ with the aggregated representation $h_{N(u)}^k$ of all its immediate neighbors $N(u)$:

$$h_u^k = \text{COMBINE}^k(h_u^{k-1}, h_{N(u)}^k) \tag{2.15}$$

This iterative computation over layers $0 \leq k \leq K$ is initialized by setting $h_u^0 = X_u$, where X_u is the node features vector.

Matrix Formulation Spatial variants of GNNs can be implemented using matrix multiplication representation, as presented in [5, 20, 51]:

$$H_{\text{agg}}^k = AH^{k-1}W_0^k \tag{2.16}$$

where $H_{\text{agg}}^k \in \mathbb{R}^{n \times d_k}$ is the tensor matrix of all aggregations $h_{N(u)}^k$, $A \in \mathbb{R}^{n \times n}$ is the adjacency matrix, $H^{k-1} \in \mathbb{R}^{n \times d_{k-1}}$ is the tensor matrix of representations h_v^{k-1} on the $(k-1)$-th layer, and $W_0^k \in \mathbb{R}^{d_{k-1} \times d_k}$ is a matrix of learnable parameters.

The combine operation is formulated as:

$$H^k = \sigma(H^k_{\text{agg}} + H^{k-1}W^k_1) \tag{2.17}$$

where $W^k_1 \in \mathbb{R}^{d_{k-1} \times d_k}$ is a second matrix of learnable parameters, and σ is a nonlinear function applied pointwise, such as ReLU [2].

Finally, after K layers, a low-dimensional node embedding $Z \in \mathbb{R}^{n \times d_K}$ is obtained as a tensor matrix, $Z = H^K$.

Types of Spatial-based GNNs In the realm of spatial-based GNNs, several variants have been proposed, each with unique characteristics and strengths. Here, a brief overview of some of the most prominent types is provided:
Graph Convolutional Networks (GCN): Introduced by Kipf and Welling [38], GCNs extend convolution operations from Euclidean space to graphs. They achieve this by aggregating the features of neighboring nodes and combining them with the features of the target node. The simplicity of the GCN has made it one of the foundational GNN models.

GraphSAGE (Graph Sample and Aggregation): Developed by Hamilton et al. [24], GraphSAGE introduced an inductive learning framework to GNNs. Unlike GCNs, which are inherently transductive and require the entire graph for training, GraphSAGE generates embeddings by sampling and aggregating features from a node's local neighborhood. This approach enables the use of node feature information even when the model is tested on unseen graphs.

Graph Attention Networks (GAT): Proposed by Veličković et al. [69], GATs incorporate attention mechanisms into GNNs. This mechanism allows GATs to assign different weights to the features of neighboring nodes during the aggregation process. As a result, GATs are capable of handling graphs with varying node degrees and complex structures.

Graph Isomorphism Network (GIN): Created by Xu et al. [77], GINs are designed to capture the graph isomorphism – the ability to discern between different graph structures. GIN uses a multilayer perceptron in its propagation model, enabling the network to learn to differentiate various graph structures.

In essence, all GNN architectures leverage the graph's topology and node features to derive informative node embeddings. However, the ways they aggregate and transform the local information differ, leading to different performances depending on the nature of the graphs and the problem at hand.

Graph Construction in DP

Graph-based representations have gained popularity in DP due to their effectiveness in capturing intricate relationships between various biological entities [3]. In these graphs, nodes and edges generally symbolize biological entities and the interactions between them, respectively. The construction of such graphs typically involves two key steps: node definition and edge definition.

Node Definition Nodes in the graph can represent different entities depending on the specific task. One common approach is to define nodes based on biologically meaningful entities. For instance, in a *cell-graph*, nodes correspond to individual cells, which can be identified either through automated methods such as deep learning models like HoverNet [23], CIA-Net [83], UNet [56], or through manual annotation by expert pathologists. Other representations, like *patch-graphs*, form nodes based on fixed-sized patches that are either randomly sampled from WSIs or selected using a method that removes non-tissue regions [33]. Moreover, nodes can also be created from meaningful tissue regions, as *cell-to-tissue graph*, identified by pre-trained deep learning models on tissue datasets (e.g., NCT-CRC-HE-100 [35]). An example of these 3 differents representation is shown in fig. 2.11.

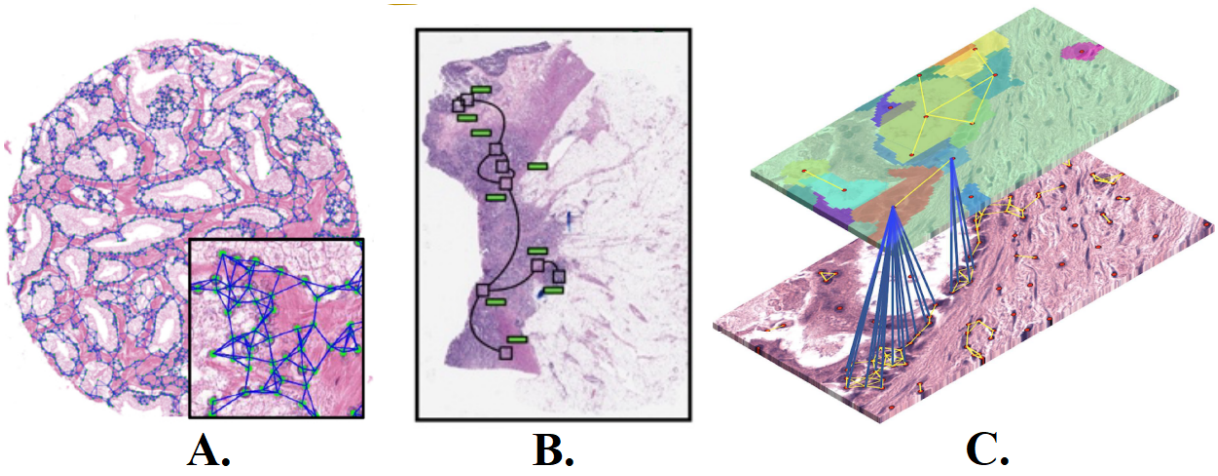


Figure 2.11: A representation of different graph-based models for histopathology images. A) Cell-graph representation for prostate cancer. B) Tissue-graph representation for colorectal cancer. C) Hierarchical cell-to-tissue graph representation for breast cancer. (Source: [3])

Edge Definition Edges within these graphs encapsulate the interactions or relationships between different biological entities. They can be defined using various methods,

each bringing their unique perspective to the interactions between entities. Proximity thresholds, nearest neighbor rules, and probabilistic models are among the commonly used techniques for edge definition. Moreover, graph construction methods like Delaunay triangulation and the nearest neighbor method further refine the process of edge definition.

Delaunay triangulation creates a set of triangles from a given set of points in a plane, ensuring that no point is inside the circumcircle of any triangle [42]. On the other hand, the nearest neighbor method connects each point to its k nearest neighbors, resulting in a k-nearest neighbor graph [55].

The combination of these edge definitions and graph construction methods enables the extraction of complex, multi-scale features from histopathological images. This unlocks new possibilities in computational pathology by forming an intricate network of relationships between biological entities [3].

2.3. Related Work

The realm of DP has witnessed remarkable advancements due to the advent and proliferation of ML techniques, particularly deep learning. These methods have been adopted to facilitate numerous tasks such as tissue segmentation, classification, and grading, leading to promising avenues of research in precision medicine and improved diagnostic techniques [3, 16, 17].

However, despite these substantial advancements, traditional CNNs and Deep Neural Networks (DNNs) can sometimes fall short in capturing the complex histological dependencies inherent in tissue images. These dependencies, which include the spatial relationships and interactions between different cell types and tissue structures, can be crucial for accurate disease diagnosis and grading.

Therefore, there is not only room but also a pressing need for the introduction of novel methodologies that can effectively capture these histological dependencies and further enhance the performance and reliability of DP tools.

The aim of this section is to provide an overview of the state-of-the-art methods employed in DP, with a focus on deep learning approaches, and GNNs in particular. The strengths and limitations of these existing methods will be discussed, and areas where the proposed solution introduces novelty and potential improvements will be identified. This discussion is crucial as it highlights the context and relevance of the work being presented and its potential contribution to this rapidly evolving field [3].

2.3.1. Deep Learning in DP

In recent years, deep learning has been leveraged extensively to make groundbreaking advancements in DP [16]. The application of deep learning has resulted in enhanced precision and automation in the analysis of various biological structures, spanning from cellular to glandular, up to region-wide scales. This review focuses on key studies in the realm of DP that have harnessed deep learning techniques to solve a variety of complex tasks.

Deep learning methodologies have been effectively utilized for cellular-level DP image analysis, including semantic segmentation of cellular objects, cell and nuclei detection, mitosis detection, and instance segmentation of cells or nuclei [16]. Different architectures such as CNNs, Fully Connected Networks (FCNs), and Stacked Autoencoders (SAEs) have been employed. Ciresan et al. [14] used an ensemble of CNNs for neuronal membranes segmentation, whereas Ronneberger et al. [56] employed a U-Net model with deformation augmentation for a similar task. Moreover, Song et al. [63] adopted a multi-scale CNN and graph partitioning-based method for cytoplasm and nuclei segmentation. For mitosis detection, a deep regression network was leveraged by Chen et al. [12].

Gland-level DP image analysis primarily employs architectures like CNNs and FCNs, often coupled with post-processing techniques for gland segmentation. Bentaieb et al. [6] implemented a multi-task learning approach for gland grading and segmentation, whereas Xu et al. [79] utilized an FCN-based semantic segmentation with a HED side convolution channel for contour detection.

At a broader scale, deep learning has made significant contributions to region-level or WSIs analysis in DP. Techniques like CNNs, FCNs, among others, have been utilized for tasks such as classification, segmentation, and region-level feature extraction. For instance, Xu et al. [78] demonstrated the application of a CNN with Multiple Instance Learning (MIL) for colon region classification and segmentation, while Wang et al. [72] used an FCN with context-aware feature selection for lung WSI classification.

However, while these techniques have achieved impressive results, they predominantly rely on pixel-level information. This dependence becomes a limiting factor as it fails to capture histological and topological relationships between biological structures. This gap presents an opportunity for novel methodologies that can encapsulate relational information between biological structures at a higher level, offering a promising direction in DP.

In the following sections, the relatively new and promising domain of graph-based deep

learning methods in digital pathology, which aims to address this limitation and has shown promising results in recent studies, are explored.

GNN Applications in DP GNNs have shown significant potential in computational pathology, demonstrating their versatility in handling complex data structures. These applications can be categorised based on the models of cellular structures used:

- **Cell-graphs:** Cell-graph models have been utilised for a number of tasks:
  - **Breast Cancer Classification:** Techniques involving cell-graphs with GINs have been developed for breast cancer classification [19, 31, 32].
  - **Prostate Cancer Classification:** Sureka et al. [66] modelled histology tissues as a graph of nuclei, employing GCNs and Random Survival Forest (RSF) for breast and prostate cancer classification.
  - **Colorectal Cancer Classification:** Cell-graphs have been used along with various GNN models for colorectal cancer classification[65, 82].
  - **Renal Cancer Classification:** Cell-graphs have been used with GraphSAGE models for renal cancer classification [13].
- **Tissue-Graphs:** Tissue-graph models have been applied for the following tasks:
  - **Breast Cancer Classification:** Lu et al. [44] utilised tissue-graphs and GIN for classifying breast cancer status.
  - **Prostate Cancer Segmentation and Classification:** Anklin et al. [4] proposed a method using a variant of GIN, SegGini, for segmentation and classification in prostate cancer.
  - **Skin Cancer Classification:** GCNs have been used for skin cancer classification [74].
- **Patch-Graphs:** Patch-graph models have been used for a number of applications:
  - **Cancer Region-of-Interest (ROI) Retrieval and Classification:** Techniques involving patch-graphs and GCNs have been employed for breast and lung cancer ROI retrieval and classification [1, 44, 53]. Notably, Ozen et al. [53] and Zheng et al. [81] proposed methodologies for breast cancer and lung cancer ROI retrieval respectively.
- **Other Models:** Some studies have used other innovative models (e.g. Cluster-Centroids-Graph[62]):

- **Cervical Cancer Classification:** Fusion CNN-GCN models have been utilised for cervical cancer classification [62].

While the majority of the studies in the literature have primarily focused on classification tasks, only a few have ventured into developing frameworks for ROI retrieval and segmentation tasks. These frameworks often employ tissue-graph or patch-graph models, underscoring the versatility and expansive potential of GNNs in digital pathology.

Nevertheless, none of these methods have touched upon the specific problem of BM identification or any closely related tasks within the context of cell-graphs. In the next section, the current works on BM identification are analyzed.

2.3.2. BM Identification

BM identification is a crucial and challenging task in digital pathology. Several studies have considered this problem [11, 70, 73]. However, most of these methods depend upon pixel level information, which falls short in capturing histological and topological relationships between the BM and various cells locally present in the tissue.

Wang et al. [70] proposed a deep learning-based method to perform the basal membrane segmentation. They regarded this as an object contour segmentation problem and modified a contour segmentation deep learning model with an adversarial training strategy.

Wu et al. [73] introduced a semi-automatic algorithm for the measurement of the glomerular basement membrane thickness in electron microscopy kidney images. This algorithm involved manually inputting a string of sparsely spaced points along the central line of the basement membrane and using two distance functions of edges to the central line to construct the membrane boundaries.

Cao et al. [11] presented a method for the automatic segmentation of the pathological glomerular basement membrane in transmission electron microscopy images. They used a random forest-based ML method, termed as random forest stacks, which can automatically segment different morphologies and gray-level basement membranes.

These studies indicate the ongoing effort to automate BM identification in digital pathology. However, the challenge remains in fully leveraging the histological and topological relationships in cell-graphs for this task. Notably, only two works have addressed this task using a cell-graph framework [27, 48]. The details of these two cornerstone studies will be discussed in the following paragraph.

GNN applications for BM identification Two seminal works [27, 48] have addressed the problem of BM identification using cell-graph frameworks, employing GNNs with a specific aim to classify edges as either crossing or non-crossing the BM.

The first work [48] uses the GraphSAGE GNN framework to learn novel node embeddings. In this approach, only node labels (cell labels) are considered for node aggregation. The classification of an edge is performed by applying a non-linear activation function to the scalar product of the two node embeddings representing the edge. While this method marks a significant advancement in GNN-based BM identification, it does not fully leverage the relational information embedded within the edges.

The second study [27] extends this concept by introducing an Edge Aggregated GNN (EAGNN) architecture that considers both node and edge features. This method introduces new features such as cell density and cell entropy for nodes, and neighborhood overlap similarity, node distance, cell entropy difference, and cell density difference for edges. The edge classifier consists of a Bidirectional Concatenation (BC) of node embeddings followed by a Multi-Layer Perceptron (MLP). This MLP serves to classify the concatenated node embeddings. The outputs are then accumulated and subjected to a sigmoid function, effectively deciding whether the edge crosses the BM or not.

Despite the novelty of these approaches in leveraging GNNs for BM identification, they still present room for improvement. In particular, these methods do not fully consider potential patterns encapsulated within triangular motifs, a critical structure in the cell-graph built using Delaunay triangulation. Furthermore, while the MLP was employed as a scoring mechanism, it lacked an optimization or training process, limiting its capacity to enhance performance iteratively. Therefore, there is still significant potential for further enhancement of these GNN-based BM identification approaches.

2.3.3. Edge Classification Task

The study of GNNs has predominantly focused on node features and their aggregation for tasks such as node and graph classification. However, this node-centric view often overlooks the potential value of edge features and the importance of context-awareness in edge classification tasks. Edge features can provide additional, sometimes crucial, information about the relationships between nodes, and a classifier that is aware of a broader context could potentially enhance the performance of edge classification.

Edge-based Aggregation While many of the commonly used GNN frameworks primarily focus on node feature aggregation layers, they may overlook the potential to fully

utilize information crucial for tasks such as edge classification. Incorporating edge features into the GNN architecture, however, is not straightforward and presents its own set of challenges. As a result, only a few works have ventured in this direction, aiming to develop methods for effective edge-based aggregation in GNNs [20, 21, 27, 76, 80].

The MPNNs model proposed in [20] aggregates edge distance features within nodes to update the embeddings. The EAGNN model [27] integrates multi-dimensional edge features with node embeddings to update node representations, which are then used for edge classification tasks in DP. In contrast, the Adaptive Edge Features Guided GAT model [21] introduces an attention mechanism that combines both node and edge features, with the edge features being adaptive across network layers, meaning that the GNN computes new edge embeddings at each update. The NENN model [80] introduces a hierarchical dual-level attention mechanism that alternately stacks node-level and edge-level attention layers to learn and aggregate embeddings for nodes and edges, allowing the node and edge embeddings to mutually reinforce each other. These works highlight the potential of edge features in enhancing the performance of GNNs. However, their direct applicability to edge classification tasks remains an open area for research.

Context-Awareness Classifier In the context of edge classification tasks within GNNs, the development of a classifier that is aware of a broader context could potentially enhance the classification performance. However, when trained on single edges, an edge classifier may lack a broader context, which could limit its performance [21]. While several recent works have proposed methods to enhance context-awareness at the GNN level, these have been primarily designed for node or graph classification tasks. For instance, the Ego-GNN model [60] exploits ego-network structures to capture both local and global structural information. Equivariant Subgraph Aggregation Networks (ESAN) [7] represent each graph as a set of subgraphs and process it using an equivariant architecture. The work on GCNs via Motif-based Attention [54] introduces a motif-matching guided subgraph normalization method to capture neighborhood information. The k-dimensional GNNs (k-GNNs) proposed in [47] take higher-order graph structures at multiple scales into account. The model proposed in [9] uses a topologically-aware message passing scheme based on substructure encoding. Lastly, MeshCNN [26], designed specifically for triangular meshes, operates on the mesh edges. However, their direct applicability to edge classification tasks remains an open area for research, and the development of context-aware techniques specifically designed for edge classification would be a valuable contribution to the field.

3 | Methodology

In this chapter, the methodology underpinning this research is laid out in detail. Beginning with foundational graph definitions, the text moves on to describe the specifics of the KI dataset. Subsequently, the architecture and functionality of the cell-graph model are elucidated, setting the stage for a comprehensive explanation of the Graph Neural Network (GNN) Framework. Within this framework, particular attention is given to the node embedding layers and the specialized edge classifier, TM-CNN. Model optimization techniques, including backpropagation and the formulation of the loss function, are discussed to provide insight into how the model was fine-tuned for optimal performance. Finally, the chapter concludes with a thorough review of the evaluation metrics employed in the study, ranging from precision and recall to F1 score, ROC-AUC Curve, and accuracy. This methodology chapter serves as a blueprint for the experiments and analyses that follow, offering a rigorous framework for evaluating the efficacy of the proposed model.

3.1. Graph definitions

Definition 3.1 (Delaunay Triangulation)**.** Given a set of vertices $\mathcal{V} = \{v_1, v_2, ..., v_n\}$ in the plane, a Delaunay triangulation $DT(\mathcal{V})$ is a triangulation such that no vertex in $\mathcal{V}$ is inside the circumcircle of any triangle in $DT(\mathcal{V})$.

Definition 3.2 (Motif)**.** Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of vertices and $\mathcal{E}$ is the set of edges, a motif $\mathcal{M} = (\mathcal{V}', \mathcal{E}')$ is a subgraph of $\mathcal{G}$, such that $\mathcal{V}' \subseteq \mathcal{V}$ and $\mathcal{E}' \subseteq \mathcal{E}$. The motif $\mathcal{M}$ represents a specific configuration of nodes and edges that recurs within the graph $\mathcal{G}$.

Definition 3.3 (Triangular Motif)**.** Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, a triangular motif $\mathcal{T} = (\mathcal{V}', \mathcal{E}')$ is a subgraph of $\mathcal{G}$, such that $\mathcal{V}' \subseteq \mathcal{V}$, $\mathcal{E}' \subseteq \mathcal{E}$, $|\mathcal{V}'| = 3$, and $|\mathcal{E}'| = 3$. All vertices in $\mathcal{V}'$ are connected to each other, forming a cycle.

Lemma 3.1 (Delaunay Triangulation Triangle Formation)**.** For a graph $\mathcal{G}$ generated by a Delaunay triangulation of a set of vertices $\mathcal{V}$ in the plane, each triangle in the triangulation is a triangular motif. Therefore, the entire graph $\mathcal{G}$ can be viewed as a collection of these

motifs.

Lemma 3.2 (Edge-Associated Triangular Motifs in Delaunay Triangulation)**.** Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a graph generated by a Delaunay triangulation of a set of vertices $\mathcal{V}$ in the plane. For each edge $(u, v) \in \mathcal{E}$, where $u, v \in \mathcal{V}$, if (u, v) is an interior edge of $\mathcal{G}$, then there exist exactly two triangular motifs in $\mathcal{G}$ that include (u, v). However, if (u, v) is a boundary edge of $\mathcal{G}$, then there exists exactly one triangular motif in $\mathcal{G}$ that includes (u, v).

The algorithm used to extract all Edge-Associated Triangular Motifs is presented in Algorithm 3.1.

Algorithm 3.1 Triangular Motif Extraction in Graphs

Require: A graph $G = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ is the set of vertices and $\mathcal{E}$ is the set of edges.
Ensure: *MOTIFS* - A set of extracted triangular motifs from the graph.

Function get_neighbors(u, v): {Return the sets of neighbors of vertices u and v}
Function get_closest_vertices(*vertices*): {Compute the two closest vertices among the common neighboring vertices}
Function PADDING(): {Perform zero-padding, or random-padding between u and v, or mirror-padding of the vertex}
Initialize *MOTIFS* as an empty set.
for each edge (u, v) in $\mathcal{E}$ **do**
    $N(u), N(v) \leftarrow$ get_neighbors(u, v)
    $vertices \leftarrow N(u) \cap N(v)$
    if $\dim(vertices) = 1$ **then**
        if DROP_EDGE **then**
            continue
        else
            $motif_1 \leftarrow (u, v, vertices[0])$
            $motif_2 \leftarrow$ PADDING()
            $MOTIFS \leftarrow MOTIFS \cup \{motif_1, motif_2\}$
        end if
    else if $\dim(vertices) = 2$ **then**
        $motif_1 \leftarrow (u, v, vertices[0])$
        $motif_2 \leftarrow (u, v, vertices[1])$
        $MOTIFS \leftarrow MOTIFS \cup \{motif_1, motif_2\}$
    else if $\dim(vertices) > 2$ **then**
        closest_vertices $\leftarrow$ get_closest_vertices(*vertices*)
        $motif_1 \leftarrow (u, v,$ closest_vertices$[0])$
        $motif_2 \leftarrow (u, v,$ closest_vertices$[1])$
        $MOTIFS \leftarrow MOTIFS \cup \{motif_1, motif_2\}$
    end if
end for
return *MOTIFS*

3.2. KI dataset

The KI dataset, sourced from the Department of Dental Medicine at Karolinska Institutet (KI), serves as the primary foundation of this study. This dataset comprises WSIs of H&E stained oral mucosa tissue gathered from patients who had received HSCT and present different levels of oral manifestations of cGVHD and healthy volunteers [68].

Due to their large size, the original WSIs were cropped into 2000 × 2000 pixel regions for easier handling. Within these images, cells have been manually labeled by a team of three experts from the Department of Dental Medicine at KI. In total, approximately 62000 cells have been manually annotated and classified [48]. Additional annotations were also performed to identify the basement membrane.

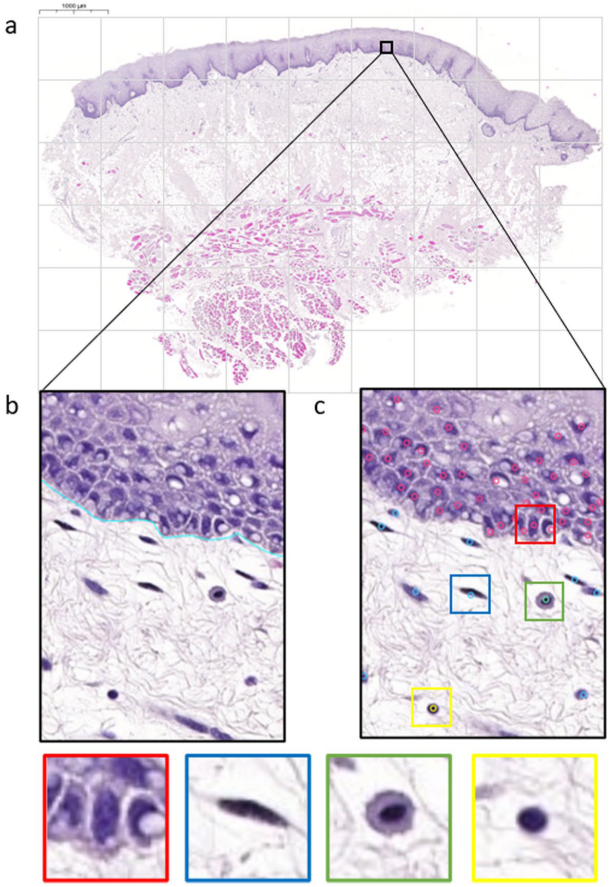


Figure 3.1: (**a**) WSI of healthy buccal oral mucosa that has been segmented into tiles of 2000 × 2000 pixels. (**b**) Annotation of the extension of the BM visualised with blue line. (**c**) Nuclei centroids were annotated and labelled as either epithelial (red), fibroblast or endothelial (blue), inflammatory (green) or lymphocytic (yellow) (Image taken from [48])

Figure 3.1 offers a visual overview of the KI dataset and its annotations.

3.3. Cell-graph model

In order to enable a comprehensive analysis of interactions and structures within tissues at the cellular level, a cell-graph model was constructed. This model utilizes tiles which incorporate BM annotations. The graph model was generated using Delaunay triangulation (see def. 3.1).

In the resulting graph, the cells served as nodes and the edges were formed based on the Delaunay triangulation. Notably, the edges were then labeled based on their interaction with the BM, classified as either crossing or not crossing the BM [27, 48].

To ensure a consistent evaluation, the same data split as in [27, 48] was employed. Thus, the cell graph dataset was divided into distinct training, validation, and testing subsets (see Table 3.1).

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

Table 3.1: Distribution of edge classes across different splits

Figure 3.2 provides a visual representation of the generated Cell-Graph models used in this study. It is crucial to note the repetitive structures linked to each edge in these models. In graphs produced using Delaunay triangulation, each edge is associated with two triangular motifs within the graph's body, and a single triangular motif on its boundaries.

The following features are considered within the dataset.

Node Features

In the graph-based model, both manually annotated and computed features for each node, which represents a cell in the original histology image, are used. The following features are used:

1. *Cell Type:* Each node $v \in V$ is labeled by its cell type $N_{\text{type}}(v) \in T$, where $T = \{$*epithelial, fibroblast and endothelial, inflammatory, lymphocyte*$\}$ is the set of cell types considered.

2. *Cell Density:* The local cell density at a node $v \in V$ is defined as the average

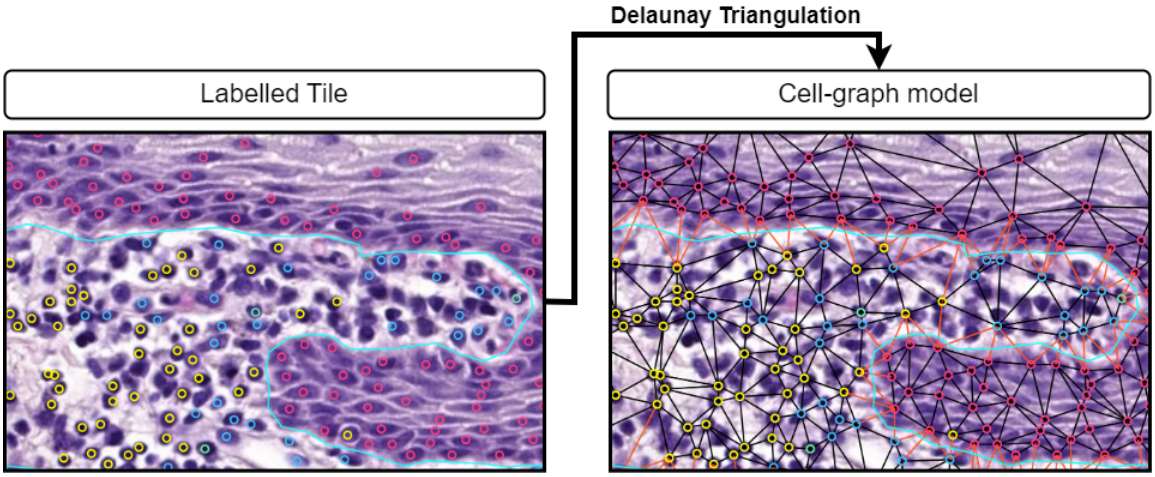


Figure 3.2: Cell-graph model generation. (Left) Manually Annotated Tile. (Right) Generated Graph; (red) BM crossing edges, (black) BM non-crossing edges.

distance between v and its immediate neighbours $u \in N(v)$.

$$N_{\text{den}}(v) = \frac{1}{|N(v)|} \sum_{u \in N(v)} d(v, u) \tag{3.1}$$

3. *Cell Entropy:* The local cell entropy of a node $v \in V$ is calculated by first finding the probability $p_v(\tau)$ of encountering a cell type $\tau \in T$ in its immediate neighbourhood. Cell entropy is then defined using the Shannon entropy measure.

$$p_v(\tau) = \frac{|u \in N(v) \text{ or } u = v : N_{\text{type}}(u) = \tau|}{|N(v)|} \tag{3.2}$$

$$N_{\text{ent}}(v) = -\sum_{\tau \in T} p_v(\tau) \times \ln p_v(\tau) \tag{3.3}$$

4. *DL Extracted Cell Features:* In addition to the above cell features, already presented in [27], a deep learning model (ResNet18 [28]), with weights pre-trained on ImageNet [39], is utilized to extract meaningful cell features N_{dnn}. This is done by cropping patches of size 32×32 around the cell from the WSIs and passing them through the model.

Edge Features

Edges in the graph represent relationships between the cells (or nodes) in the histological image. For each edge, the following labels are used:

1. *Node Distance:* The Euclidean distance between the endpoints of each edge $u, v \in \mathcal{E}$:

$$\hat{E}^{\text{dis}}_{u,v} = d(u, v) \tag{3.4}$$

2. *Cell Density Difference:* The difference in cell density along edge $u, v \in \mathcal{E}$:

$$\hat{E}^{\text{den}}_{u,v} = N_{\text{den}}[v] - N_{\text{den}}[u] \tag{3.5}$$

3. *Cell Entropy Difference:* The difference in cell entropy along an edge $u, v \in \mathcal{E}$:

$$\hat{E}^{\text{ent}}_{u,v} = |N_{\text{ent}}[v] - N_{\text{ent}}[u]| \tag{3.6}$$

4. *Neighbourhood Overlap Similarity:* The Sorenson similarity index quantifies the overlap between the neighbourhoods of two nodes u and v.

$$\hat{E}^{\text{nei}}_{u,v} = \frac{2|N(u) \cap N(v)|}{\deg(u) + \deg(v)} \tag{3.7}$$

5. *BM Crossing:* A binary valued BM crossing measure for each edge $u, v \in \mathcal{E}$: $\hat{E}^{\text{bm}}_{u,v} = 1$ if u, v crosses the BM, otherwise $\hat{E}^{\text{bm}}_{u,v} = 0$.

6. *DL Extracted Edge Features:* Following the same line of approach used for node features, deep learning edge features E_{dnn} are also extracted using a pre-trained ResNet-18 [28]. Considering the edges vary in size, patches of size 128×128 are cropped around the edge's center from the WSIs and passed through the model to derive these features.

The feature values are normalized to maintain the feature scale during the message passing operations in the GNN layer. The normalization[1] is defined for feature $1 \leq p \leq P$ and edge $(u, v) \in \mathcal{E}$ as:

$$E^{X}_{uvp} = \frac{\hat{E}^{X}_{uvp}}{\sum_{v' \in N(u)} |\hat{E}^{X}_{uv'p}|} \tag{3.8}$$

3.4. GNN Framework

In this section, the GNN framework is presented. The framework consists of two primary components: Node Embedding Layers and an Edge Classifier. The discussion initially

[1]Normalization computed as presented in [27]

focuses on the Node Embedding Layers, a crucial component of the architecture that is rooted in the methodology proposed by EAGNN from Nair et al. [27]. The generation of rich node embeddings is accomplished by this component.

The discourse then shifts to the novel edge classifier, the Triangular Motifs Convolutional Neural Network (TM-CNN). In contrast to traditional approaches [27, 48], the TM-CNN is explicitly engineered to extract significant patterns pertinent to edge classification tasks. A comprehensive perspective GNN Framework will be presented throughout this section.

3.4.1. Node Embedding Layers

The proposed architecture builds upon the node embedding layers proposed in [27], EAGNN.

The node embedding layers aim to derive latent node representations from a graph. The EAGNN architecture follows a layered structure, where each layer k computes a low-dimensional (d_k-dimensional) representation $h_u^k \in \mathbb{R}^{d_k}$ of the graph structure around node u.

In the EAGNN, the computation at each layer k is slightly different from the general GNN. Instead of performing an AGGREGATE operation on nodes followed by a COMBINE operation, the EAGNN incorporates multiple edge features by aggregating edge features into the node embeddings. The major difference between the EAGNN layer and other GNN layers is the way that edge features are normalized before aggregation (equation 3.8).

Following the matrix multiplication as in Equation 2.16, the aggregation operation of the proposed model at layer k, named EAgg^k, is formulated as follows:

$$\text{EAgg}^k(E_p, H^{k-1}) = E_p H^{k-1} W_0^k \tag{3.9}$$

where E_p is the p-th feature matrix of the edges (i.e. $E_p \in \mathbb{R}^{n \times n}$ is the projection of $E \in \mathbb{R}^{n \times n \times P}$ edge feature tensor onto the single edge feature p).

Then, the previous node representation is combined using the combine operation formulated in Equation 2.17. These aggregation and combining operations are performed for each edge feature, and they are then concatenated all together. Therefore, the formula for the k-th EAGNN layer is given by:

$$H^k = \sigma[\|_{p=1}^{P} (E_p H^{k-1} W_0^k + H^{k-1} W_1^k)] \tag{3.10}$$

where $\|$ denotes the concatenation operator and σ a non-linear function. Note that this non-linear function is not used in the final layer K of the node embedding layers. After K embedding layers, the node representation z_u is given by $z_u = H_u^K$ for node u.

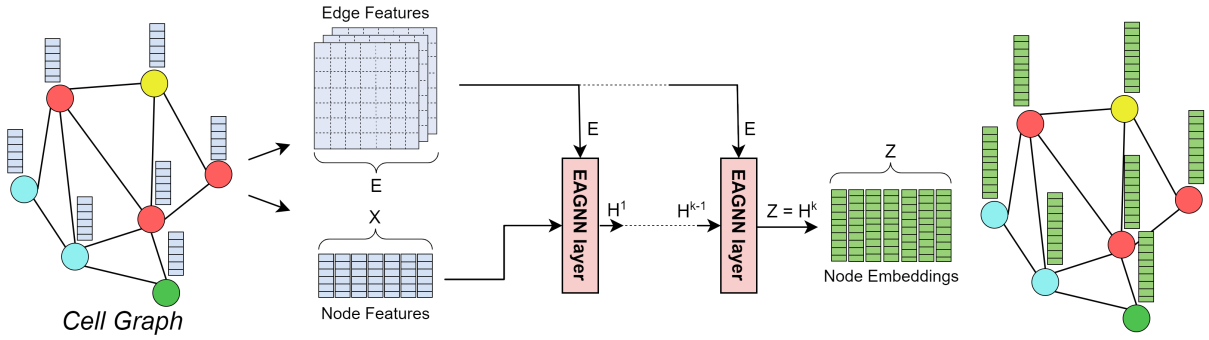


Figure 3.3: EAGNN architecture overview.

3.4.2. Edge Classifier: TM-CNN

Contrasting with traditional approaches that employ MLPs or multiplication operators [27, 48], the Triangular Motifs Convolutional Neural Network (TM-CNN) is proposed as a uniquely tailored edge classifier, designed specifically for discerning meaningful patterns pertinent to edge classification tasks.

The TM-CNN framework for edge classification is inspired by methodologies commonly used in pattern learning tasks such as Text Classification [36, 45, 71] and Financial Time-Series Classification [41]. These tasks often involve generating text or time-series embeddings and passing them through 1D-Convolutional layers. This approach allows for the extraction of local and global features, which are crucial for accurate classification. Similarly, in this work, the TM-CNN classifier aims to extract and learn meaningful patterns from the motif structures for effective edge classification.

The TM-CNN classifier is designed to take as input the two triangular motifs associated with an edge (u, v). These triangular motifs are formed by the nodes (u, v, z) and (u, v, w), where u, v, z, and w represent the node embeddings learned after K-iteration by the EAGNN architecture. Each node embedding has a size of $\mathbb{R}^{d_K}$, resulting in an input to the TM-CNN classifier of size $d_k \times 4$ (corresponding to 4 channels).

The TM-CNN architecture consists of two 1D convolutional layers and two fully connected (FC) layers:

1. The first convolutional layer takes an input of size $(d_k \times 4)$ and applies n_{f_1} filters. The output F_1 of this layer is a feature map of size $(d_k \times n_{f_1})$, where each feature is a result

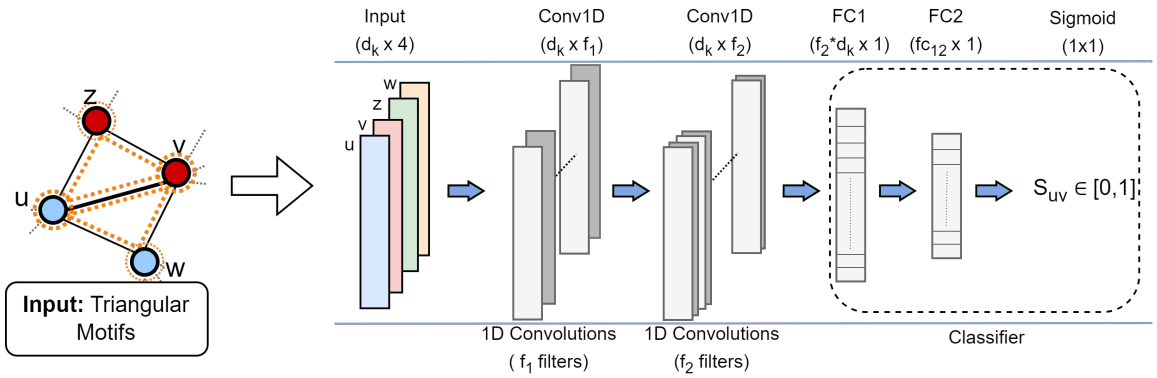


Figure 3.4: Triangular Motifs CNN (TM-CNN) architecture overview.

of the convolution operation applied to the input and the filter. Mathematically, this operation can be represented as:

$$F_{1,j} = \text{Conv1D}(I, f_{1,j}) \forall j = 1, \ldots, n_{f_1} \tag{3.11}$$

$$F_1 = [F_{1,1}|F_{1,2}|\ldots|F_{1,n_{f_1}}] \tag{3.12}$$

where I is the input (i.e. the 4 node embeddings), $f_{1,j}$ is the j-th filter, $F_{1,j}$ is the j-th vector of the feature map and F_1 is the feature map.

2. The second convolutional layer takes the feature map F_1 as input and applies n_{f_2} filters. The output of this layer is a feature map of size $(d_k \times n_{f_2})$. This operation can be represented as:

$$F_{2,j} = \text{Conv1D}(F_1, f_{2,j}) \forall j = 1, \ldots, n_{f_2} \tag{3.13}$$

$$F_2 = [F_{2,1}|F_{2,2}|\ldots|F_{2,n_{f_2}}] \tag{3.14}$$

Following the convolutional layers[2], the feature map F_2 is flattened into a vector of size $(d_k * n_{f_2})$ and passed through two FC layers:

1. The first FC layer (FC1) takes the flattened feature vector as input and applies a linear transformation followed by a non-linear activation function σ. The output size of this layer is $n_{fc_{12}}$. This operation can be represented as:

$$FC1 = \text{ReLU}(W_1 \cdot F_2 + b_1) \tag{3.15}$$

[2]Each convulotional layer employs a ReLU [2] activation function for non-linearity

where W_1 and b_1 are the weight matrix and bias vector of the FC layer, respectively.

2. The second fully connected layer (FC2) receives the output of FC1, carrying out another linear transformation. However, in this case, the output is directly passed through a sigmoid function, resulting in a scalar value between 0 and 1. The output size of this layer is therefore 1. This operation can be described as:

$$s_{uv} = \sigma(W_2 \cdot FC1 + b_2) \tag{3.16}$$

where $s_{uv} \in [0, 1]$ represents the edge classification output of the TM-CNN classifier for an edge (u, v) and σ the sigmoid activation function. This output value is then compared to a predetermined threshold to make a final binary classification (either 0 or 1).

Figure 3.4 gives an overview of the proposed architecture.

Padding

Edges located on the boundary have a single triangular motif associated with them. To ensure that the input to the TM-CNN remains consistent for both inner graph edges and boundary edges, it's necessary to apply a Padding technique. This is an essential step in filling any missing parts of the TM-CNN input. The following three distinct strategies have been proposed:

Zero-Padding: In this method, the absent portions of the triangular motif are filled with zeros. Despite being straightforward and computationally efficient, this technique may fail to accurately mirror the information that might exist in the missing sections.

Random-Padding: This strategy populates the missing sections by generating the absent triangle, with the edge (u, v) serving as the base, and either u or v selected randomly as the vertex. Although this method introduces variability to the input, it could also potentially incorporate noise, thereby misdirecting the model.

Mirror-Padding: This approach involves mirroring the existing values within the triangular motif to complete the missing sections. As it relies on the existing values from the same motif, it might offer a more accurate representation of the absent information. However, it could also inadvertently amplify any present noise or errors within the data.

3.5. Model Optimization and Backpropagation

The training of the proposed model involves the use of a specific loss function and an optimization algorithm, coupled with the backpropagation method for computing gradients.

3.5.1. Loss Function and Optimization

The Binary Cross-Entropy (BCE) loss function is employed, designed for binary classification tasks, and is defined as:

$$L(s_{uv}, y_{uv}) = -y_{uv} \log(s_{uv}) - (1 - y_{uv}) \log(1 - s_{uv}) \tag{3.17}$$

where s_{uv} is the edge classification output of the TM-CNN classifier for an edge (u, v), and y_{uv} denotes the actual label for the edge.

Optimization of the model's parameters is achieved through the Adaptive Moment Estimation (Adam) optimizer [37]. In Adam, each weight and bias in the model maintains separate learning rates, which are adaptively adjusted over the course of training. Updates of the weights and bias are performed using the formulas:

$$W_{new} = W_{old} - \alpha \frac{\partial L}{\partial W_{old}} \tag{3.18}$$

$$b_{new} = b_{old} - \alpha \frac{\partial L}{\partial b_{old}} \tag{3.19}$$

where α is the learning rate and L represents the loss function.

3.5.2. Backpropagation

Backpropagation is utilized to efficiently compute the gradient of the loss function with respect to the model parameters. It comprises two main steps:

1. *Forward Pass*: The input is propagated through the network to compute the output. The computed output is then used to evaluate the loss function.
2. *Backward Pass*: The gradient of the loss function is computed with respect to the model's parameters. This is achieved by propagating the gradient of the loss function backwards through the network.

The gradient of the loss function with respect to the parameters at the i-th layer, denoted as θ_i, is given by:

$$\frac{\partial L}{\partial \theta_i} = \frac{\partial L}{\partial a_i} \cdot \frac{\partial a_i}{\partial \theta_i} \tag{3.20}$$

where a_i is the output of the i-th layer. This gradient is computed as:

$$\frac{\partial L}{\partial a_i} = \frac{\partial L}{\partial a_{i+1}} \cdot \frac{\partial a_{i+1}}{\partial a_i} \tag{3.21}$$

where a_{i+1} is the output of the $(i+1)$-th layer.

In the proposed model architecture, the input is passed through the EAGNN to compute node embeddings during the forward pass. These embeddings combined, form triangular motifs which are input to the TM-CNN to compute the edge classification output. The BCE loss function computes the loss between the predicted output and the actual labels.

During the backward pass, the gradient of the loss is propagated backwards through the TM-CNN layers and then through the EAGNN layers to update the model parameters. Unlike previous methodologies [27, 48], the error is backpropagated through the entire structure, enhancing the accuracy of edge classification. The Adam optimizer performs the parameter updates based on these computed gradients.

3.6. Evaluation Metrics

Several evaluation metrics have been employed to assess the performance of the proposed model. These metrics provide different perspectives on the model's ability to correctly classify instances, and together they offer a comprehensive view of the model's performance.

3.6.1. Precision

Precision, also known as the positive predictive value, measures the proportion of true positive predictions (i.e., correctly identified positive instances) among all instances that the model predicted as positive. It is calculated as follows:

$$Precision = \frac{TP}{TP + FP} \tag{3.22}$$

where TP is the number of true positives and FP is the number of false positives. A high precision indicates a low rate of false-positive errors.

3.6.2. Recall

Recall, also known as sensitivity or true positive rate, measures the proportion of true positive predictions among all actual positive instances. It is calculated as follows:

$$Recall = \frac{TP}{TP + FN} \tag{3.23}$$

where FN is the number of false negatives. A high recall indicates a low rate of false-negative errors.

3.6.3. F1 score

The F1 score is the harmonic mean of precision and recall, providing a single metric that balances the two values. Unlike the arithmetic mean, the harmonic mean tends towards the smaller of the two values, so a high F1 score requires both precision and recall to be high. The F1 score is calculated as follows:

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{3.24}$$

3.6.4. ROC-AUC Curve

The Receiver Operating Characteristic (ROC) curve is a plot that illustrates the performance of a binary classifier as the discrimination threshold is varied. The curve is created by plotting the true positive rate (recall) against the false positive rate (1 - specificity) at various threshold settings. The Area Under the ROC Curve (AUC) provides a scalar measure of the model's performance across all thresholds, with a value of 1.0 representing perfect classification and 0.5 representing a model that performs no better than random chance.

3.6.5. Accuracy

Accuracy is the most intuitive performance measure. It is simply the ratio of correctly predicted instances to the total instances in the dataset. It is calculated as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.25}$$

where TN represents true negatives. However, accuracy can be misleading in the presence of class imbalance, as it may simply reflect the underlying class distribution. Therefore, it is important to consider accuracy alongside other metrics such as precision, recall, and the F1-score.

4 | Experiments and Results

In this chapter, a comprehensive empirical analysis of the proposed models for assessing basement membrane (BM) conditions is presented. Through a mixture of quantitative and qualitative evaluations, the capabilities, advantages, and limitations of the models are explored. Methodologies for training and testing are delineated, and performance metrics are employed to offer both a general and detailed understanding of model efficacy.

4.1. Experimental Setup

The cell-graph dataset was divided into two subsets: 70% used for training and 30% allocated for testing. A portion of the training dataset, 15%, was reserved for validation. The population sizes of the edge classes in the dataset, as shown in Table 3.1, indicate a data imbalance towards BM non-crossing edges.

The GNN models, denoted by EAGNN$_i$ where i represents the number of layers used in the model, along with TM-CNN, were trained using backpropagation to minimize the mean of the binary cross-entropy loss function across each mini-batch.

The main GNN configuration (EAGNN$_i$,TM-CNN) used is presented in Table 4.1.

Training was performed for 100 epochs with a mini-batch size of 32. Each batch included 16 BM crossing edges and 16 BM non-crossing edges to avoid bias derived from the imbalance in the data.

The Adam optimizer was used with an initial learning rate of 0.001. The learning rate was scheduled to decrease by 0.1 after every 40 epochs. To prevent overfitting on the training data, dropout with a probability of 0.5 was applied to EAGNN$_i$ layers and the TM-CNN classifier. Furthermore, the weight-decay parameter was set to 0.0001.

This configuration will be used throughout the following sections for all experiments. For any additional configurations, refer to Appendix A

EAGNN$_i$						
# layers	Nodes	Edges	Hidden Dim	Output Dim	Activation	
i	dim(NF)	dim(EF)	64	64	ELU [15]	
TM-CNN						
# channels	Channel Dim	Filters 1	Filters 2	Activation	FC1	FC2
4	64	16	32	ReLU[2]	$(32 \times 64, 128)$	$(128, 1)$

Table 4.1: Model Configuration. (EAGNN$_i$) NF and EF represent respectively node and edge features considered in the study, and their dimension determined respectively the Node and the Edge Feature Matrices input dimensions.

Computational Resources The experiments were performed on a workstation with the following hardware and software specifications:

- CPU: AMD Ryzen 5 5600-X 6-core
- GPU: NVIDIA GeForce RTX 3060
- RAM: 16GB
- Operating System: Windows 10 Education N

For software, the following versions were used:

- Python: 3.8
- PyTorch: 1.7.1

These specifications provide context for the computational cost and feasibility of the proposed methods. This information is also crucial for other researchers who might want to replicate the experiments or compare them with other methods.

4.2. Model Evaluation

Model evaluation encompasses two essential phases: validation and testing. These stages are indispensable for optimizing the model configuration and subsequently testing its efficiency in handling novel data.

During the validation phase, the various model configurations are assessed on the validation data. This process aids in identifying the configuration that delivers the optimal results and fine-tuning its parameters for maximum efficacy.

Once the model is optimized through validation, it is subjected to the testing phase. In

this stage, the performance of the model on unseen data is evaluated and juxtaposed with two baseline methods previously proposed in the literature [27, 48]. This comparison provides a measure of the model's ability and effectiveness in practical scenarios, offering an unbiased insight into its real-world performance.

4.2.1. Validation Results

Different configurations of the (EAGNN$_i$,TM-CNN) model were trained, and their performance was subsequently evaluated on the validation set. This step was crucial as it enabled us to assess the model's ability to generalize to unseen data, and if necessary, adjust hyperparameters.

A critical hyperparameter in this context is the threshold parameter τ. This threshold is used to adjust the score s_{uv}, essentially dictating whether it is classified as 0 or 1. The tuning of this threshold was conducted with respect to the F1 score metric on the validation set.

Table 4.2, displaying the performance metrics (including precision, recall, F1 score, ROC-AUC curve, and accuracy) for each (EAGNN$_i$,TM-CNN) configuration on the validation set, was employed to ascertain the optimal number of aggregation layers within the EAGNN backbone model.

Precision, recall and F1 score are computed with respect to the minority class, BM crossing edges.

After conducting an ablation study on both Node Features (NF) and Edge Features (EF) as detailed in Appendix A and as per the insights from [27], only N_1 and E_{1234} features were retained for the final model. The features N_{234} and E_5 did not contribute to an improvement in the model's performance metrics and were thus excluded from the final feature set[1].

The inclusion of additional information into node embeddings via aggregation, rather than limiting aggregation to two EAGNN layers, as proposed in [27], has demonstrated substantial benefits. As evidenced in Table 4.2, extending the aggregation process to five layers has led to a notable improvement in the F1 score for BM crossing edges. However, it is worth mentioning that in terms of accuracy and ROC-AUC curve, all configurations demonstrated similar performance due to the significant imbalance favoring non-crossing

[1] N_1 corresponds to Node Type (N_{type}), N_2 to Cell Density (N_{den}), N_3 to Cell Entropy (N_{ent}) and N_4 to DL Extracted Cell fearutes (N_{dnn}). Similarly, E_1 is Node Distance ($\hat{E}^{dis}$), E_2 is Cell Density difference ($\hat{E}^{den}$), E_3 is Cell Entropy difference ($\hat{E}^{ent}$), E_4 is Neighborhood Overlap Similarity ($\hat{E}^{nei}$) and E_5 is DL Extracted Edge Features ($\hat{E}_{dnn}$).

Model	Classifier	NF	EF	Precision	Recall	F1	ROC-AUC	Accuracy
$EAGNN_2$	TM-CNN	N_1	E_{1234}	0.8792	0.8529	0.8658	**0.9930**	0.9854
$EAGNN_3$	TM-CNN	N_1	E_{1234}	0.8873	0.8554	0.8711	0.9926	0.9860
$EAGNN_4$	TM-CNN	N_1	E_{1234}	0.8882	0.8563	0.8720	0.9927	0.9861
$EAGNN_5$	TM-CNN	N_1	E_{1234}	**0.9018**	0.8640	**0.8825**	0.9926	**0.9873**
$EAGNN_6$	TM-CNN	N_1	E_{1234}	0.8846	**0.8657**	0.8751	**0.9930**	0.9864

Table 4.2: Performance metrics of the various ($EAGNN_i$,TM-CNN) configurations on the validation set. The metrics are employed to ascertain the optimal number of aggregation layers (i) within the EAGNN backbone model. Node Features (NF) and Edge Features (EF) represent aggregated node and edge characteristics, respectively.

($EAGNN_5$, TM-CNN)	Precision	Recall	F1
$\tau = 0.97$	0.8915	**0.8717**	0.8815
$\tau =$ **0.98**	0.9018	0.8640	**0.8825**
$\tau = 0.99$	**0.9157**	0.8358	0.8739

Table 4.3: Precision, Recall and F1 score metrics of the ($EAGNN_5$,TM-CNN) configuration with varying threshold τ on the validation set. The metrics are used to determine the optimal value for the threshold τ.

edges.

The selection of the optimal threshold value, denoted as $\hat{\tau}$, was meticulously executed by tuning it on the validation set. As displayed in Table 4.3, the choice of $\hat{\tau}$ significantly influences the precision, recall, and F1 score metrics. The table reports the metrics as computed by the ($EAGNN_5$, TM-CNN) model configuration. The optimal threshold $\hat{\tau}$ was determined to be 0.98, which maximizes the F1 score with respect to BM crossing edges on the validation set.

With regard to the classifier architecture, various configurations of convolutional layer sizes and activation function were evaluated. However, no discernible improvement was observed in these experiments. Detailed information regarding these variations can be found in Appendix A (refer to Table A.1 for specifics).

4.2.2. Testing Results

In the testing phase of the experiment, the performance of the proposed models was compared with two models previously reported in the literature. Specifically, the ($EAGNN_2$, BC+MLP) model presented by Nair et al. in [27] and the (GraphSAGE, MUL) model from [48].

Model	Classifier	NF	EF	Precision	Recall	F1	ROC-AUC	Accuracy
[48] GraphSAGE	MUL	N_1	-	0.3417	0.7011	0.4594	-	0.9071
[27] $EAGNN_2$	BC+MLP	N_1	E_{1234}	**0.8548**	0.8427	0.8395	0.9799	0.9822
$EAGNN_2$	TM-CNN	N_1	E_{1234}	0.8523	0.8595	0.8559	0.9866	0.9837
$EAGNN_5$	TM-CNN	N_1	E_{1234}	0.8526	**0.8657**	**0.8591**	**0.9874**	**0.9840**

Table 4.4: Benchmarking of GNN models, edge classifiers, and aggregated node and edge features for BM prediction on the oral mucosa cell-graph dataset.

In these comparisons, BC stands for bidirectional concatenation, which consists in combining the embeddings of two nodes through concatenation in both directions to obtain a final score in an orientation-invariant way. Meanwhile, MUL refers to a multiplication operation between the embeddings of two nodes, generating a score to classify the edge.

The comparative results provide a valuable reference point, allowing us to assess the performance improvements offered by the proposed models in the context of established approaches in the field. They facilitate a comprehensive evaluation of the effectiveness of the proposed models in accurately predicting BM crossing edges.

As exhibited in Table 4.4, all proposed experiment configurations surpass the performance of both ($EAGNN_2$,BC+MLP) [27] and (GraphSage, MUL) [48]. As the ($EAGNN_2$,BC+MLP) [27] model had already proven superior to the (GraphSage, MUL) [48] model, it has been chosen as the primary benchmark for further discussion. The proposed model configuration, ($EAGNN_5$,TM-CNN), demonstrates a significant enhancement, with an F1 score on BM crossing edges that is approximately 2% higher. Maintaining a similar level of precision while exhibiting superior recall, meaning that fewer false negatives are yielded by this model.

4.3. Qualitative Results

In this section, a qualitative analysis of the results produced by the ($EAGNN_5$,TM-CNN) model, identified as the best in this study, is conducted. For comparison, the ($EAGNN_2$,BC+MLP) model from [27] is also included. The BM prediction results are depicted in Figures 4.1 and 4.2, offering a visual representation of the respective performance of each model.

Figure 4.1 features the predictions made by both models on an intact BM sample from the test set. The actual BM annotation is represented by the blue line, with true positive, false negative, and false positive edge predictions represented by green, yellow, and red lines respectively. A similar performance level is observed for both models in this scenario,

showcasing their comparable capability in classifying BM crossing edges in healthy BM samples.

Figure 4.2 shows a sample presenting a degraded BM. The superior performance of the (EAGNN$_5$,TM-CNN) model becomes more noticeable under these more challenging conditions. Even with the complications introduced by the degraded BM, BM crossing edges are identified with increased accuracy by the (EAGNN$_5$,TM-CNN) model.

This improved performance contributes to an approximately 10% increase in the F1 score metrics related to BM crossing edges. A significant reduction in the number of false positive predictions is achieved by the (EAGNN$_5$,TM-CNN) model, particularly evident in the decreased noise within the epithelial layer, as depicted in Figure 4.2(b). This reduction in error exemplifies the robustness of the model and its ability to maintain high performance even in challenging circumstances.

This qualitative analysis provides a visual confirmation of the statistical results previously presented, demonstrating the tangible improvements offered by the (EAGNN$_5$,TM-CNN) model. The potential of the model for real-world applications, where accurate BM prediction is of utmost importance, is emphasised.

4.3.1. Graph Segmentation

To intuitively visualize the model's predictions, a color-coded segmentation was employed. This process assigns colors to the triangular motifs, recurring in the graph, based on the number of BM crossing edges per triangle.

The segmentation categories are as follows[2]:

(Empty) Triangles with 0 BM crossing edges

(Red) Triangles with only 1 BM crossing edge

(Yellow) Triangles with 2 BM crossing edges

(Green) Triangles with 3 BM crossing edges

Red triangles indicate areas where the BM is compromised, indicating breaks or disruptions in its structure. In contrast, yellow and green triangles denote regions where the BM remains contiguous and intact. For further details, please refer to Appendix B.

The efficacy of the proposed (EAGNN$_5$,TM-CNN) model in segmenting oral tissue graphs

[2]It should be noted that cases with n BM crossing edges, where $n > 3$, have been excluded from consideration

is effectively elucidated through visual examples. In Figures 4.3 and 4.4, a comparative perspective is offered between the ground truth and the segmentations generated by the ($EAGNN_2$,BC+MLP)[27] and the ($EAGNN_5$,TM-CNN) models.

The segmentation of a healthy oral tissue graph is depicted in Figure 4.3. The ground truth is displayed in the leftmost panel for comparison, the segmentations derived from the ($EAGNN_2$,BC+MLP)[27] model are exhibited in the middle panel, and the results produced by the ($EAGNN_5$,TM-CNN) model are shown in the rightmost panel.

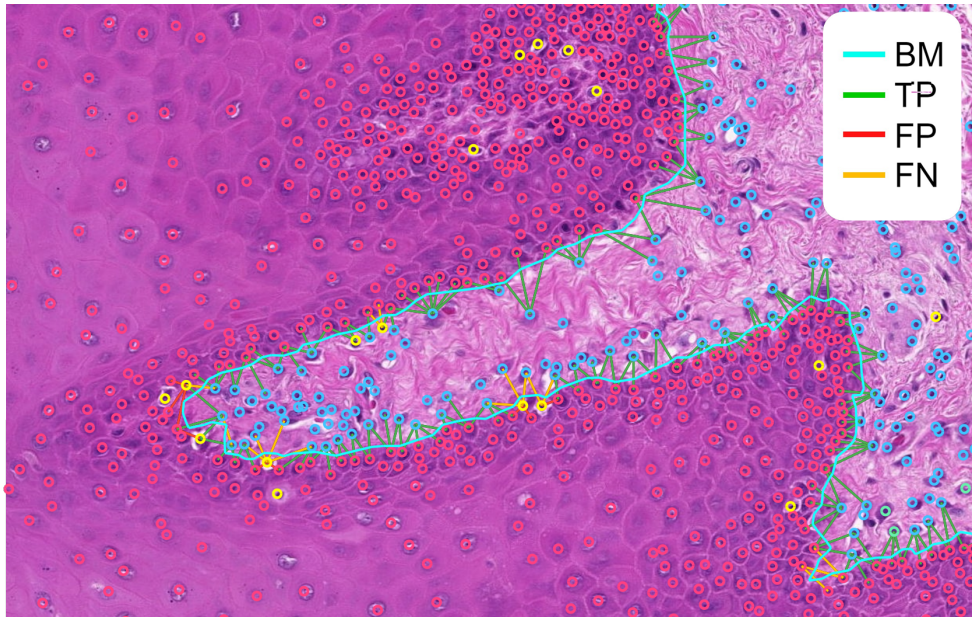
The segmentation for an oral tissue graph featuring BM breakages is displayed in Figure 4.4. The panels from left to right represent the ground truth, the outcomes from the ($EAGNN_2$,BC+MLP)[27] model, and the segmentations achieved by the ($EAGNN_5$,TM-CNN) model.

In the outcomes achieved by the ($EAGNN_5$,TM-CNN) model, a notable reduction in noisy predictions is observed, significantly enhancing the clarity of the segmentation. This represents an improvement over the results yielded by the ($EAGNN_2$,BC+MLP)[27] model, where noisy predictions were more prevalent. Particularly within the epithelial layer, the ($EAGNN_5$,TM-CNN) model has been able to accurately identify regions that were previously misclassified as BM segments by the ($EAGNN_2$,BC+MLP)[27] model. This leads to a more accurate depiction of BM segments.

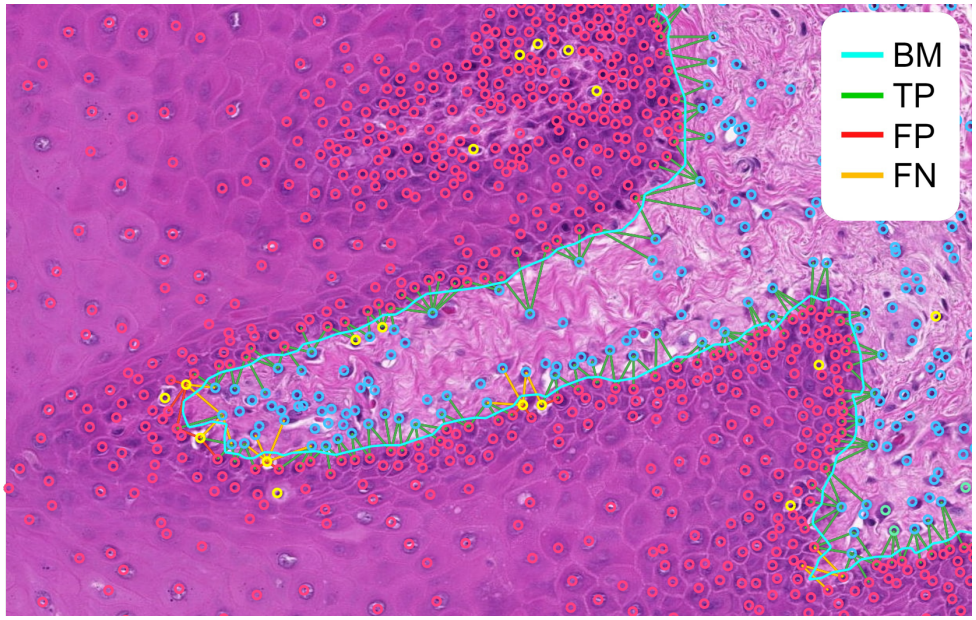
Moreover, the preservation of the BM structure by the ($EAGNN_5$,TM-CNN) model is apparent, even though it isn't flawless. Conversely, the application of the ($EAGNN_2$,BC+MLP) model results in a loss of information along the BM, accompanied by the occurrence of false predictions far from the BM as depicted in Figure 4.4.

In addition, the ($EAGNN_5$,TM-CNN) model considerably reduces the number of red triangles, which signify potential BM breakages. This reduction in the portrayal of BM breakages is particularly conspicuous in the example provided in Figure 4.4. Thus, a more accurate and clearer visual representation of the tissue's condition is provided by the ($EAGNN_5$,TM-CNN) model.

These visual portrayals allow for an immediate comparison between the segmentations produced by the two models and the ground truth, highlighting the relative effectiveness of each model.

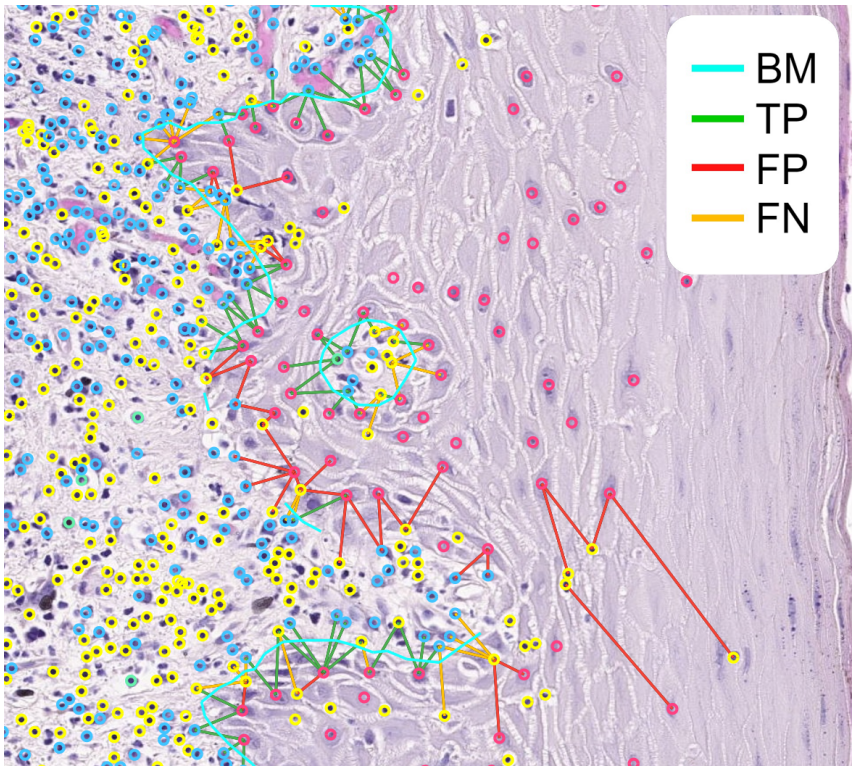


(a) (EAGNN$_2$,BC+MLP)[27] F1 score: 0.9297 (BM crossing edges)

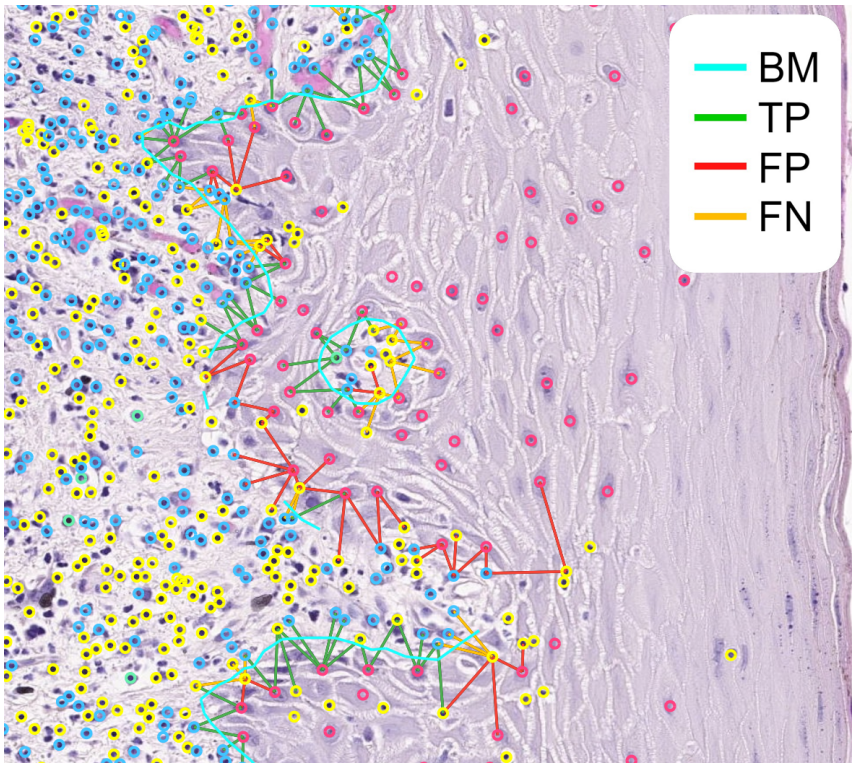


(b) (EAGNN$_5$,TM-CNN) F1 score: 0.9286 (BM crossing edges)

Figure 4.1: Comparison of (EAGNN$_2$,BC+MLP) [27] and (EAGNN$_5$,TM-CNN) predictions for healthy BM tissue sample. The blue line is the BM annotation. The green, yellow and red lines represent true positive, false negative and false positive edge predictions respectively

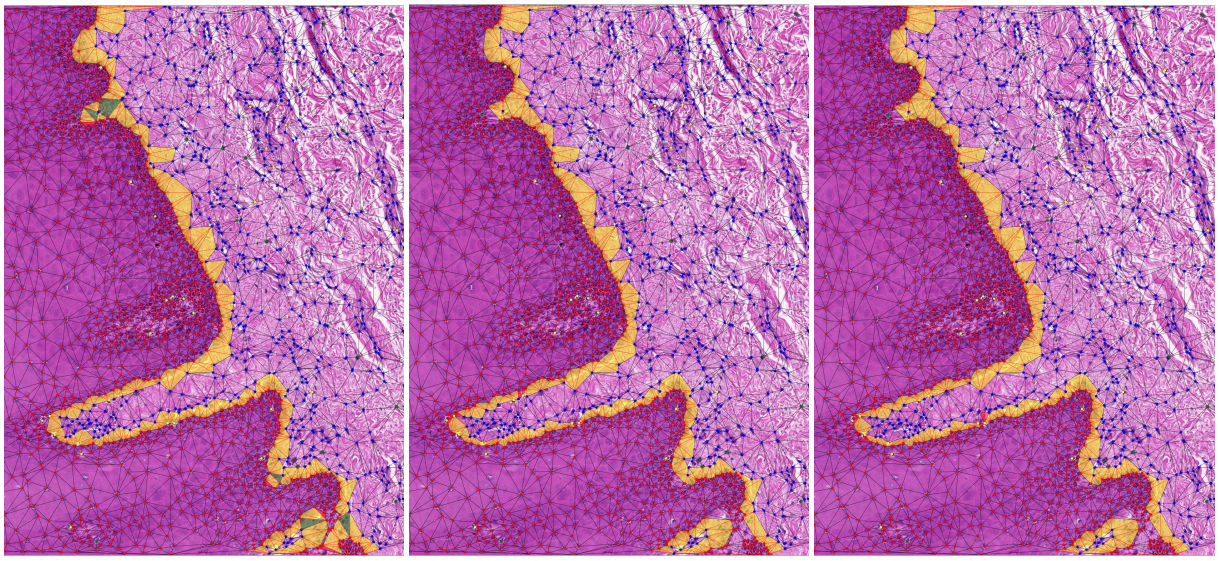


(a) (EAGNN$_2$,BC+MLP)[27] F1 score: 0.5329 (BM crossing edges)



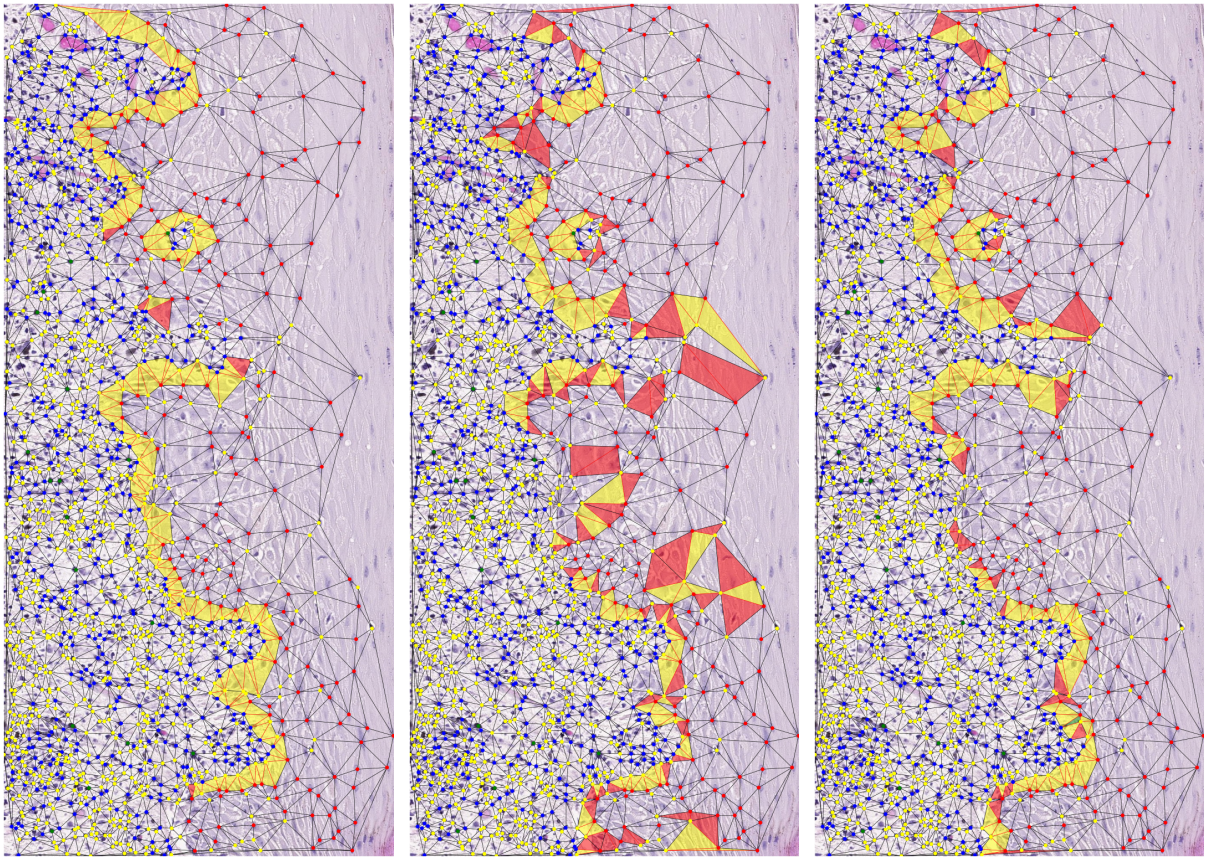
(b) (EAGNN$_5$,TM-CNN) F1 score: 0.6340 (BM crossing edges)

Figure 4.2: Comparison of (EAGNN$_2$,BC+MLP) [27] and (EAGNN$_5$,TM-CNN) predictions for degraded BM tissue sample. The blue line is the BM annotation. The green, yellow and red lines represent true positive, false negative and false positive edge predictions respectively



(a) Ground Truth (b) (EAGNN$_2$,BC+MLP)[27] (c) (EAGNN$_5$,TM-CNN)

Figure 4.3: Segmentation of an healthy Oral Tissue's graph. Red triangles indicate disruptions in the BM structure, while yellow and green triangles indicate BM continuity.



(a) Ground Truth (b) (EAGNN$_2$,BC+MLP)[27] (c) (EAGNN$_5$,TM-CNN)

Figure 4.4: Segmentation of an Oral Tissue's graph presenting BM breakages. Red triangles indicate disruptions in the BM structure, while yellow and green triangles indicate BM continuity.

5 | Discussion

The problem of accurately identifying and localizing basement membranes (BMs) and their breakages in oral tissue samples was addressed in this research. This issue holds significant relevance in the field of computational pathology, especially for pathologists studying oral cGVHD and histopathological severity grading [68]. The methodology adopted involved the development and implementation of a novel model, ($EAGNN_5$,TM-CNN), for edge classification tasks in cell-graph models. Its performance was compared against an existing model, ($EAGNN_2$,BC+MLP), as proposed by Nair et al. [27].

The findings obtained from this study have been promising. Notably, superior effectiveness in identifying BMs, particularly in samples with degraded BMs, was demonstrated by the ($EAGNN_5$,TM-CNN) model. This enhanced performance is largely a result of the error being backpropagated through the entire structure, leading to the optimized training of both the GNN and the classifier, as well as the inclusion of convolutional layers to capture common combinations of recurrent structures within the graph. Furthermore, the interpretability of the model was enhanced through the introduction of a new graph segmentation technique in this study, thus proving beneficial for pathologists.

In this discussion section, the key findings of the study, their implications, and potential areas for further research will be explored. Additionally, some of the limitations of the current study will be discussed and future directions proposed to overcome these limitations and enhance the utility of the model.

5.1. Interpretations

The effectiveness of the proposed model, ($EAGNN_5$,TM-CNN), is attributed to an amalgamation of several factors. First and foremost, the extended training implemented at both the GNN level and the classifier level allowed for a more nuanced understanding of BM structures and their aberrations. This comprehensive training played a pivotal role in the superior recognition capabilities of the model, particularly in samples containing degraded BMs.

An integral aspect of the model's performance comes from integrating triangular motifs into the edge classification task. As expected, these recurrent structures, inherent in a Delaunay generated graph, provided a broader context for classifying edges, which in turn led to improved performance. This strategic integration validated the hypothesis that accounting for common patterns in the data structure would enhance the model's predictive capabilities.

The incorporation of convolutional layers in the classifier model served as an additional enhancement. These layers enabled the model to further recognize and capture patterns in triangular motifs, reinforcing the model's ability to get a picture of the broader context of cell-BM interactions. Consequently, the model's ability to detect and classify edges, as either BM crossing or non-crossing, accurately was considerably boosted.

New features (i.e., N_{dnn} and E_{dnn}) for the GNN node embeddings layer were introduced, building upon the work of [27, 48]. Despite these additions, the performance of the proposed model was not improved, as evidenced in the conducted ablation study (see Appendix A). This lack of enhancement could be directly attributed to the pre-training of the feature extraction model on ImageNet [39], rather than a dataset more specific to the task.

In addition to these enhancements, the introduction of a new graph segmentation technique proved instrumental in refining the model's interpretability. This technique facilitated the partitioning of the graph into distinct segments, simplifying the interpretation of results. This, in turn, is particularly beneficial for pathologists, as it allowed for a clearer understanding of the segmented graph results compared to edge results. Consequently, it enabled a better appreciation of the model's performance in detecting and localizing BMs and their breakages in oral tissue samples.

In summary, the combination of extended training, convolutional layers, the consideration of triangular motifs, and the novel graph segmentation technique led to the superior performance and enhanced interpretability of the (EAGNN$_5$,TM-CNN) model. This not only furthered the model's edge classification capabilities but also opened new avenues for understanding and interpreting cell-BM interactions in cell-graph models.

5.2. Implications

The results obtained from this study hold significant implications for both the field of computational pathology and clinical practice. The ability of the (EAGNN$_5$,TM-CNN) model to accurately identify and localize BMs, especially in samples with degraded BM,

can greatly assist pathologists in their work. By improving diagnostic accuracy, this model could potentially enhance patient outcomes, especially for those suffering from conditions such as oral cGVHD where early and accurate detection is vital [68].

Moreover, the enhanced interpretability offered by the novel graph segmentation technique provides pathologists with a nuanced understanding of BM conditions. This interpretability serves not primarily to streamline the diagnosis, but to offer robust decision support that can mitigate observer variance. By facilitating a deeper and more consistent understanding of the tissue's condition, our model can be a valuable tool in guiding both the assessment of disease severity and the subsequent course of treatment, potentially leading to more targeted and effective treatment strategies.

In addition, the integration of triangular motifs into the edge classification task has broader implications. It highlights the importance of considering the inherent structure of the data in computational models. This approach has the potential to extend beyond computational pathology and may be valuable for a wide range of graph-based tasks, where accounting for structural context could yield improved results.

Furthermore, the enhanced performance and interpretability of the (EAGNN_5,TM-CNN) model may also encourage the broader application of computational models in pathology. It demonstrates the potential benefits of integrating advanced computational techniques into traditional pathological processes, thereby paving the way for future advancements in computational pathology.

On a broader level, this research also provides a new perspective for understanding and interpreting cell-BM interactions. The unique approach of using graph-based models to represent and analyze tissue samples could open up new avenues for research, potentially leading to the discovery of new biological insights and diagnostic techniques. Consequently, the implications of this study extend beyond the immediate goal of improving the identification and localization of BMs, potentially influencing the wider field of computational biology.

5.3. Limitations

An attempt was made in this study to enhance the model's performance by incorporating new features, different from those employed by Nair et al. [27]. However, as evidenced in the ablation study (refer to Tables A.2 and A.3 in Appendix A), these modifications did not yield the expected improvement in performance. A plausible explanation for this outcome might be associated with the nature of the deep learning features used, which

were extracted using a ResNet18 [28] model pretrained on the ImageNet [39] dataset. The pretrained model, although effective in many contexts, might not be entirely apt for capturing the specific features relevant in tissue images, thereby limiting its effectiveness in this application.

Furthermore, it was noted that the model currently lacks richer edge representations which could provide a more granular understanding of cellular interactions. The development of these intricate edge representations, while complex, could be an avenue for future research to further enhance the model's capabilities.

As a future direction, it would be of interest to explore the use of models pretrained on datasets more closely related to the task at hand. More specifically, employing a model pretrained on tissue image datasets might offer a richer and more relevant set of features for the task, potentially enhancing the model's performance (as it has been presented in [3]). Further research is needed to investigate this possibility and its potential impact on the model's performance.

5.4. Future Works

In terms of future research and developments, this study has identified several promising directions. The primary focus of this work has been on the identification and localization of BMs and their breakages, employing the (EAGNN$_5$,TM-CNN) model which has demonstrated superior performance over existing approaches. However, there remain numerous avenues for refinement and enhancement of the model as well as its potential applications.

While the (EAGNN$_5$,TM-CNN) model already presents a substantial improvement in handling BM identification, a natural extension of this work could be to augment the model to also grade the severity of BM breakages. This advancement could lead to the development of a more comprehensive tool that not only assists pathologists in the identification of BM conditions but also aids in determining their severity, which is a crucial aspect of disease diagnosis and treatment planning [68].

While the results of this study show promise for the (EAGNN$_5$,TM-CNN) model, there is always room for future refinement and exploration. This study has primarily focused on improving the classifier within the GNN framework for edge classification. However, one potential area of improvement could be the node and edge representations. The EAGNN architecture, while able to aggregate edge features into node embeddings, may lack robust edge representations. This could be addressed by enhancing the edge classification task to consider not only node embeddings but also edge embeddings.

In this vein, there have been some novel approaches to improving edge representations. For instance, [21] proposed a framework for updating edge representation at each aggregation step using an attention mechanism. This method, however, has been criticized by Yang et al. [80] as not being suitable for edge or link prediction tasks due to its linear edge representation update, which they suggest can lead to a loss of edge information.

As an alternative, Yang et al. proposed a framework using a dual-attention mechanism for node and edge embeddings updates, called the Node and Edge features in Graph Neural Network (NENN). The NENN consists of a node-level attention layer and an edge-level attention layer that are alternately stacked to learn and aggregate embeddings for nodes and edges. This approach allows the node and edge embeddings to mutually reinforce each other, providing a potentially more effective method of edge information update.

As a future direction, it could be fruitful to implement NENN within the GNN framework and evaluate its effectiveness in the BM identification task. While NENN has been shown to outperform other methods in academic citations and molecular graphs, its application in a digital pathology context has not yet been explored. The introduction of such a methodology could further enhance the performance of the model, potentially leading to even more accurate identification of BMs in oral tissue samples.

In addition to grading the severity of BM breakages, the ($EAGNN_5$,TM-CNN) model could be potentially extended and adapted for a variety of tasks, such as tissue segmentation, tumor region segmentation, boundary detection, and region of interest (ROI) retrieval. Furthermore, it could also be employed for classification tasks.

The value of such an approach is evident when considering the work of other researchers in this field. For instance, Anklin et al. [4] proposed a tissue-graph model for prostate cancer segmentation using GIN. Moreover, patch-graph approaches have been used for retrieving regions of interest in breast and lung cancers [1, 44, 53]. The proposed model could potentially enhance the performance of these tasks by focusing on the cellular level, as opposed to lower level structures such as patches and tissue graphs.

By expanding the application of the ($EAGNN_5$,TM-CNN) model to these tasks, it could serve as a versatile tool in the realm of computational pathology. This opens up a range of opportunities for further research and development, potentially leading to breakthroughs in the diagnosis and treatment of various diseases. The proposed model could be a valuable addition to the existing repertoire of AI-powered tools in the field of pathology, complementing and enhancing the capabilities of current approaches.

6 | Conclusions

This research presented a novel method for the identification and localization of basement membranes (BMs) in oral tissue samples, a significant challenge in computational pathology, particularly concerning oral cGVHD and tissue severity grading [68]. Unlike most existing methods in the literature for edge classification or link prediction, which primarily depend on embeddings of two individual nodes, the proposed model, the (EAGNN$_5$,TM-CNN), has integrated motifs associated with edges to achieve improved performance in edge classification. This innovative approach has not only demonstrated superior performance but also helped reduce noise in predictions, effectively eliminating isolated prediction cases that are typically unlikely to occur in real scenarios.

The proposed model was evaluated against the (EAGNN$_2$,BC+MLP) model suggested by Nair et al. [27], and the results revealed a marked improvement, especially in the identification of degraded BMs. The enhanced performance can be attributed to the extended training at the GNN level and classifier level and the incorporation of convolutional layers to capture recurrent structures within the graph. Additionally, the introduction of a novel graph segmentation technique significantly improved the interpretability of the model, a crucial aspect for real-world applicability.

While the results were promising, the study was not without limitations. Despite the incorporation of new features, the performance of the model did not significantly improve, which can be partly attributed to the pretraining of the feature extraction model on ImageNet [39], a less task-specific dataset. This has highlighted the importance of using more relevant pretraining datasets, particularly those specific to tissue images.

The current study sets the stage for numerous avenues of future research. Potential areas include adapting the model to grade the severity of BM breakages, investigating the impact of using models pretrained on more relevant datasets, and enhancing the model's edge representations. Furthermore, the model could be trained on a larger dataset, thereby enhancing its robustness and reducing sensitivity to data imbalance. Such improvements could significantly enhance the model's generalization ability and reliability.

Moreover, the broad applicability of the (EAGNN$_5$,TM-CNN) model suggests its po-

tential use in a variety of computational pathology tasks. This potential to contribute significantly to the diagnosis and treatment of various diseases underscores the model's significance.

In conclusion, this research has made an important contribution to computational pathology by developing an innovative model that effectively identifies and localizes BMs in oral tissue samples. By providing enhanced interpretability of results, the model can be more easily utilized by pathologists. Nonetheless, the research also emphasizes the need for ongoing investigation and refinement to address identified limitations and fully unlock the model's potential.

Bibliography

[1] M. Adnan, S. Kalra, and H. R. Tizhoosh. Representation learning of histopathology images using graph neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 988–989, 2020.

[2] A. F. Agarap. Deep learning using rectified linear units (relu). *CoRR*, abs/1803.08375, 2018. URL http://arxiv.org/abs/1803.08375.

[3] D. Ahmedt-Aristizabal, M. A. Armin, S. Denman, C. Fookes, and L. Petersson. A survey on graph-based deep learning for computational histopathology. *CoRR*, abs/2107.00272, 2021. URL https://arxiv.org/abs/2107.00272.

[4] V. Anklin, P. Pati, G. Jaume, B. Bozorgtabar, A. Foncubierta-Rodríguez, J.-P. Thiran, M. Sibony, M. Gabrani, and O. Goksel. Learning whole-slide segmentation from inexact and incomplete labels using tissue graphs, 2021.

[5] J. Atwood and D. Towsley. Search-convolutional neural networks. *CoRR*, abs/1511.02136, 2015. URL http://arxiv.org/abs/1511.02136.

[6] A. Bentaieb, J. Kawahara, and G. Hamarneh. Multi-loss convolutional networks for gland analysis in microscopy. pages 642–645, 04 2016. doi: 10.1109/ISBI.2016.7493349.

[7] B. Bevilacqua, F. Frasca, D. Lim, B. Srinivasan, C. Cai, G. Balamurugan, M. M. Bronstein, and H. Maron. Equivariant subgraph aggregation networks. *CoRR*, abs/2110.02910, 2021. URL https://arxiv.org/abs/2110.02910.

[8] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006. URL /bib/bishop/Bishop2006/Pattern-Recognition-and-Machine-Learning-Christophe-M-Bishop.pdf,/bib/bishop/Bishop2006/978-0-387-31073-2_sm.pdf,https://www.microsoft.com/en-us/research/people/cmbishop/#!prml-book.

[9] G. Bouritsas, F. Frasca, S. Zafeiriou, and M. M. Bronstein. Improving graph neural

network expressivity via subgraph isomorphism counting. *CoRR*, abs/2006.09252, 2020. URL https://arxiv.org/abs/2006.09252.

[10] M. M. Bronstein, J. Bruna, T. Cohen, and P. Velickovic. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *CoRR*, abs/2104.13478, 2021. URL https://arxiv.org/abs/2104.13478.

[11] L. Cao, Y. Lu, C. Li, and W. Yang. Automatic segmentation of pathological glomerular basement membrane in transmission electron microscopy images with random forest stacks. *Computational and mathematical methods in medicine*, 2019:1684218, 03 2019. doi: 10.1155/2019/1684218.

[12] H. Chen, X. Wang, and P. Heng. Automated mitosis detection with deep regression networks. pages 1204–1207, 04 2016. doi: 10.1109/ISBI.2016.7493482.

[13] R. J. Chen, M. Y. Lu, J. Wang, D. F. Williamson, S. J. Rodig, N. I. Lindeman, and F. Mahmood. Pathomic fusion: an integrated framework for fusing histopathology and genomic features for cancer diagnosis and prognosis. *IEEE Transactions on Medical Imaging*, 2020.

[14] D. Ciresan, A. Giusti, L. Gambardella, and J. Schmidhuber. Deep neural networks segment neuronal membranes in electron microscopy images. In F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper_files/paper/2012/file/459a4ddcb586f24efd9395aa7662bc7c-Paper.pdf.

[15] D. Clevert, T. Unterthiner, and S. Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). In Y. Bengio and Y. LeCun, editors, *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016. URL http://arxiv.org/abs/1511.07289.

[16] S. Deng, X. Zhang, W. Yan, E. I.-C. Chang, Y. Fan, M. Lai, and Y. Xu. Deep learning in digital pathology image analysis: a survey. *Frontiers of Medicine*, 14(4): 470, 2020. doi: 10.1007/s11684-020-0782-9. URL https://journal.hep.com.cn/fmd/EN/abstract/article_27600.shtml.

[17] A. Esteva, A. Robicquet, B. Ramsundar, V. Kuleshov, M. DePristo, K. Chou, C. Cui, G. Corrado, S. Thrun, and J. Dean. A guide to deep learning in healthcare. *Nature Medicine*, 25, 01 2019. doi: 10.1038/s41591-018-0316-z.

[18] W. K. Funkhouser. Chapter 11 - pathology: The clinical description of human disease. In W. B. Coleman and G. J. Tsongalis, editors, *Molecular Pathology*, pages 197–207. Academic Press, San Diego, 2009. ISBN 978-0-12-374419-7. doi: https://doi.org/10.1016/B978-0-12-374419-7.00011-1. URL https://www.sciencedirect.com/science/article/pii/B9780123744197000111.

[19] S. Gadiya, D. Anand, and A. Sethi. Histographs: Graphs in histopathology, 2019.

[20] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl. Neural message passing for quantum chemistry. *CoRR*, abs/1704.01212, 2017. URL http://arxiv.org/abs/1704.01212.

[21] L. Gong and Q. Cheng. Adaptive edge features guided graph attention networks. *CoRR*, abs/1809.02709, 2018. URL http://arxiv.org/abs/1809.02709.

[22] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. URL http://www.deeplearningbook.org.

[23] S. Graham, Q. D. Vu, S. E. A. Raza, J. T. Kwak, and N. M. Rajpoot. XY network for nuclear segmentation in multi-tissue histology images. *CoRR*, abs/1812.06499, 2018. URL http://arxiv.org/abs/1812.06499.

[24] W. L. Hamilton, R. Ying, and J. Leskovec. Inductive representation learning on large graphs. *CoRR*, abs/1706.02216, 2017. URL http://arxiv.org/abs/1706.02216.

[25] M. G. Hanna and L. Pantanowitz. Digital pathology. In R. Narayan, editor, *Encyclopedia of Biomedical Engineering*, pages 524–532. Elsevier, Oxford, 2019. ISBN 978-0-12-805144-3. doi: https://doi.org/10.1016/B978-0-12-801238-3.99958-6. URL https://www.sciencedirect.com/science/article/pii/B9780128012383999586.

[26] R. Hanocka, A. Hertz, N. Fish, R. Giryes, S. Fleishman, and D. Cohen-Or. Meshcnn: A network with an edge. *CoRR*, abs/1809.05910, 2018. URL http://arxiv.org/abs/1809.05910.

[27] T. Hasegawa, H. Arvidsson, N. Tudzarovski, K. Meinke, R. V. Sugars, and A. Ashok Nair. Edge-based graph neural networks for cell-graph modeling and prediction. In *Information Processing in Medical Imaging: 28th International Conference, IPMI 2023, San Carlos de Bariloche, Argentina, June 18–23, 2023, Proceedings*, page 265–277, Berlin, Heidelberg, 2023. Springer-Verlag. ISBN 978-3-031-34047-5. doi: 10.1007/978-3-031-34048-2_21. URL https://doi.org/10.1007/978-3-031-34048-2_21.

[28] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. URL http://arxiv.org/abs/1512.03385.

[29] A. E. Hoerl and R. W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970. doi: 10.1080/00401706.1970.10488634. URL https://www.tandfonline.com/doi/abs/10.1080/00401706.1970.10488634.

[30] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015. URL http://arxiv.org/abs/1502.03167. cite arxiv:1502.03167.

[31] G. Jaume, P. Pati, B. Bozorgtabar, A. Foncubierta-Rodríguez, F. Feroce, A. M. Anniciello, T. Rau, J. Thiran, M. Gabrani, and O. Goksel. Quantifying explainers of graph neural networks in computational pathology. *CoRR*, abs/2011.12646, 2020. URL https://arxiv.org/abs/2011.12646.

[32] G. Jaume, P. Pati, A. Foncubierta-Rodríguez, F. Feroce, G. Scognamiglio, A. M. Anniciello, J. Thiran, O. Goksel, and M. Gabrani. Towards explainable graph representations in digital pathology. *CoRR*, abs/2007.00311, 2020. URL https://arxiv.org/abs/2007.00311.

[33] S. Kalra, H. Tizhoosh, C. Choi, S. Shah, P. Diamandis, C. J. Campbell, and L. Pantanowitz. Yottixel – an image search engine for large archives of histopathology whole slide images. *Medical Image Analysis*, 65:101757, 2020. ISSN 1361-8415. doi: https://doi.org/10.1016/j.media.2020.101757. URL https://www.sciencedirect.com/science/article/pii/S1361841520301213.

[34] K. J. Kaplan and L. K. Rao, editors. *Digital Pathology: Historical Perspectives, Current Concepts & Future Applications*. Medicine, Medicine (R0). Springer Cham, 1 edition, 2016. ISBN 978-3-319-20378-2. doi: 10.1007/978-3-319-20379-9. URL https://doi.org/10.1007/978-3-319-20379-9.

[35] J. N. Kather, N. Halama, and A. Marx. 100,000 histological images of human colorectal cancer and healthy tissue, Apr. 2018. URL https://doi.org/10.5281/zenodo.1214456.

[36] H. Kim and Y.-S. Jeong. Sentiment classification using convolutional neural networks. *Applied Sciences*, 9(11):2347, Jun 2019. ISSN 2076-3417. doi: 10.3390/app9112347. URL http://dx.doi.org/10.3390/app9112347.

[37] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In Y. Bengio

and Y. LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

[38] T. N. Kipf and M. Welling. Semi-supervised classification with graph convolutional networks. *CoRR*, abs/1609.02907, 2016. URL http://arxiv.org/abs/1609.02907.

[39] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012. URL http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.

[40] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *nature*, 521(7553):436, 2015.

[41] L. Liu and Y. W. Si. 1d convolutional neural networks for chart pattern classification in financial time series. *The Journal of Supercomputing*, 78, 08 2022. doi: 10.1007/s11227-022-04431-5.

[42] Y. Liu and G. Yin. The delaunay triangulation learner and its ensembles. *Computational Statistics & Data Analysis*, 152:107030, 2020. ISSN 0167-9473. doi: https://doi.org/10.1016/j.csda.2020.107030. URL https://www.sciencedirect.com/science/article/pii/S0167947320301213.

[43] Z. Liu and J. Zhou. Introduction to graph neural networks. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 2020.

[44] W. Lu, S. Graham, M. Bilal, N. Rajpoot, and F. Minhas. Capturing cellular topology in multi-gigapixel pathology images. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1049–1058, 2020. doi: 10.1109/CVPRW50498.2020.00138.

[45] E. Merdivan, A. Vafeiadis, D. Kalatzis, S. Hanke, J. Kroph, K. Votis, D. Giakoumis, D. Tzovaras, L. Chen, R. Hamzaoui, and M. Geist. Image-based text classification using 2d convolutional neural networks. pages 144–149, 2019. doi: 10.1109/SmartWorld-UIC-ATC-SCALCOM-IOP-SCI.2019.00066.

[46] T. M. Mitchell. *Machine learning*, volume 1. McGraw-hill New York, 1997.

[47] C. Morris, M. Ritzert, M. Fey, W. L. Hamilton, J. E. Lenssen, G. Rattan, and M. Grohe. Weisfeiler and leman go neural: Higher-order graph neural networks. *CoRR*, abs/1810.02244, 2018. URL http://arxiv.org/abs/1810.02244.

[48] A. Nair, H. Arvidsson, J. E. G. V., N. Tudzarovski, K. Meinke, and R. V. Sugars. A graph neural network framework for mapping histological topology in oral mucosal tissue. *BMC Bioinformatics*, 23(1):506, 2022. ISSN 1471-2105. doi: 10.1186/s12859-022-05063-5. URL https://doi.org/10.1186/s12859-022-05063-5.

[49] A. Nanci. Chapter 1 - structure of the oral tissues. In A. Nanci, editor, *Ten Cate's Oral Histology (Eighth Edition)*, pages 1–13. Mosby, St. Louis (MO), eighth edition edition, 2013. ISBN 978-0-323-07846-7. doi: https://doi.org/10.1016/B978-0-323-07846-7.00001-X. URL https://www.sciencedirect.com/science/article/pii/B978032307846700001X.

[50] A. Nanci. Chapter 12 - oral mucosa. In A. Nanci, editor, *Ten Cate's Oral Histology (Eighth Edition)*, pages 278–310. Mosby, St. Louis (MO), eighth edition edition, 2013. ISBN 978-0-323-07846-7. doi: https://doi.org/10.1016/B978-0-323-07846-7.00012-4. URL https://www.sciencedirect.com/science/article/pii/B9780323078467000124.

[51] M. Niepert, M. Ahmed, and K. Kutzkov. Learning convolutional neural networks for graphs. *CoRR*, abs/1605.05273, 2016. URL http://arxiv.org/abs/1605.05273.

[52] T. E. of Encyclopaedia Britannica. histology. https://www.britannica.com/science/histology, 2013. Accessed 17 April 2023.

[53] Y. Ozen, S. Aksoy, K. Kösemehmetoğlu, S. Önder, and A. Üner. Self-supervised learning with graph neural networks for region of interest retrieval in histopathology. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 6329–6334, 2021. doi: 10.1109/ICPR48806.2021.9412903.

[54] H. Peng, J. Li, Q. Gong, Y. Ning, and L. Wang. Graph convolutional neural networks via motif-based attention. *CoRR*, abs/1811.08270, 2018. URL http://arxiv.org/abs/1811.08270.

[55] F. P. Preparata and M. I. Shamos. *Computational Geometry - An Introduction*. Texts and Monographs in Computer Science. Springer, 1985. ISBN 3-540-96131-3. doi: 10.1007/978-1-4612-1098-6. URL https://doi.org/10.1007/978-1-4612-1098-6.

[56] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015. URL http://arxiv.org/abs/1505.04597.

[57] F. Rosenblatt. The perceptron: A probabilistic model for information storage and

organization in the brain. *Psychological Review*, 65(6):386–408, 1958. ISSN 19391471. doi: 10.1037/h0042519.

[58] M. H. Ross and W. Pawlina. *Histology : a text and atlas : with correlated cell and molecular biology*. Philadelphia : Wolters Kluwer/Lippincott Williams & Wilkins Health, 6th ed. edition, 2011. ISBN 0781772001. URL http://lib.ugent.be/catalog/rug01:001638312.

[59] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[60] D. Sandfelder, P. Vijayan, and W. L. Hamilton. Ego-gnns: Exploiting ego structures in graph neural networks. *CoRR*, abs/2107.10957, 2021. URL https://arxiv.org/abs/2107.10957.

[61] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, Jan 2009. ISSN 1941-0093. doi: 10.1109/TNN.2008.2005605.

[62] J. Shi, R. Wang, Y. Zheng, Z. Jiang, and L. Yu. Graph convolutional networks for cervical cell classification. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2019*, 2019.

[63] Y. Song, L. Zhang, S. Chen, D. Ni, B. Lei, and T. Wang. Accurate segmentation of cervical cytoplasm and nuclei based on multi-scale convolutional network and graph partitioning. *IEEE transactions on bio-medical engineering*, 62, 05 2015. doi: 10.1109/TBME.2015.2430895.

[64] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014. URL http://www.cs.toronto.edu/~rsalakhu/papers/srivastava14a.pdf.

[65] 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)*, pages 3636–3643, 2021. doi: 10.1109/ICPR48806.2021.9412535.

[66] M. Sureka, A. Patil, D. Anand, and A. Sethi. Visualization for histopathology images using graph convolutional neural networks. In *2020 IEEE 20th International Conference on Bioinformatics and Bioengineering (BIBE)*. IEEE, oct 2020. doi: 10.

1109/bibe50027.2020.00060. URL https://doi.org/10.1109%2Fbibe50027.2020.00060.

[67] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society (Series B)*, 58:267–288, 1996.

[68] V. Tollemar, N. Tudzarovski, G. Warfvinge, N. Yarom, M. Remberger, R. Heymann, K. Garming Legert, and R. V. Sugars. Histopathological grading of oral mucosal chronic graft-versus-host disease: Large cohort analysis. *Biology of Blood and Marrow Transplantation*, 26(10):1971–1979, 2020. ISSN 1083-8791. doi: https://doi.org/10.1016/j.bbmt.2020.06.031. URL https://www.sciencedirect.com/science/article/pii/S1083879120304018.

[69] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio. Graph attention networks. In *ICLR 2018*, 2017. URL http://arxiv.org/abs/1710.10903.

[70] D. Wang, C. Gu, K. Wu, and X. Guan. Adversarial neural networks for basal membrane segmentation of microinvasive cervix carcinoma in histopathology images. In *2017 International Conference on Machine Learning and Cybernetics (ICMLC)*, volume 2, pages 385–389, 2017. doi: 10.1109/ICMLC.2017.8108952.

[71] R. Wang, Z. Li, J. Cao, T. Chen, and L. Wang. Convolutional recurrent neural networks for text classification. pages 1–6, 2019. doi: 10.1109/IJCNN.2019.8852406.

[72] X. Wang, H. Chen, C. Gan, H. Lin, Q. Dou, Q. Huang, M. Cai, and P.-A. Heng. Weakly supervised learning for whole slide lung cancer image classification. In *Medical Imaging with Deep Learning*, 2018. URL https://openreview.net/forum?id=SJwod1hjz.

[73] H.-S. Wu, S. Dikman, and J. Gil. A semi-automatic algorithm for measurement of basement membrane thickness in kidneys in electron microscopy images. *Computer Methods and Programs in Biomedicine*, 97(3):223–231, 2010. ISSN 0169-2607. doi: https://doi.org/10.1016/j.cmpb.2009.07.002. URL https://www.sciencedirect.com/science/article/pii/S0169260709002107.

[74] J. Wu, J.-X. Zhong, E. Z. Chen, J. Zhang, J. Y. Jay, and L. Yu. Weakly and semi-supervised graph cnn for identifying basal cell carcinoma on pathological images. In *Graph Learning in Medical Imaging*, pages 112–119. Springer, 2019.

[75] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu. A comprehensive survey on graph neural networks. *CoRR*, abs/1901.00596, 2019. doi: 10.1109/TNNLS.2020.2978386. URL http://arxiv.org/abs/1901.00596.

[76] T. Xie and J. C. Grossman. Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties. *Phys. Rev. Lett.*, 120: 145301, Apr 2018. doi: 10.1103/PhysRevLett.120.145301. URL https://link.aps.org/doi/10.1103/PhysRevLett.120.145301.

[77] K. Xu, W. Hu, J. Leskovec, and S. Jegelka. How powerful are graph neural networks? *CoRR*, abs/1810.00826, 2018. URL http://arxiv.org/abs/1810.00826.

[78] Y. Xu, T. Mo, Q. Feng, P. Zhong, M. Lai, and E. I.-C. Chang. Deep learning of feature representation with multiple instance learning for medical image analysis. *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1626–1630, 2014.

[79] Y. Xu, Y. Li, Y. Wang, M. Liu, Y. Fan, M. Lai, and E. I. Chang. Gland instance segmentation using deep multichannel neural networks. *CoRR*, abs/1611.06661, 2016. URL http://arxiv.org/abs/1611.06661.

[80] Y. Yang and D. Li. Nenn: Incorporate node and edge features in graph neural networks. In S. J. Pan and M. Sugiyama, editors, *Proceedings of The 12th Asian Conference on Machine Learning*, volume 129 of *Proceedings of Machine Learning Research*, pages 593–608. PMLR, 18–20 Nov 2020. URL https://proceedings.mlr.press/v129/yang20a.html.

[81] Y. Zheng, B. Jiang, J. Shi, H. Zhang, and F. Xie. Encoding histopathological wsis using gnn for scalable diagnostically relevant regions retrieval. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2019*, pages 550–558, 2019.

[82] Y. Zhou, S. Graham, N. A. Koohbanani, M. Shaban, P.-A. Heng, and N. Rajpoot. Cgc-net: Cell graph convolutional network for grading of colorectal cancer histology images, 2019.

[83] Y. Zhou, O. F. Onder, Q. Dou, E. Tsougenis, H. Chen, and P. Heng. Cia-net: Robust nuclei instance segmentation with contour-aware information aggregation. *CoRR*, abs/1903.05358, 2019. URL http://arxiv.org/abs/1903.05358.

A | Appendix A

A.1. Model Selection

This appendix section provides a summary of performance metrics for different filter configurations of the TM-CNN model combined with the EAGNN$_5$ backbone. For these configurations, the node features (NF) and edge features (EF) were set as N_1 and E_{1234}, respectively. The performance metrics were evaluated on the test set.

Table A.1 presents the precision, recall, and F1 score for three different configurations of the TM-CNN model, where n_{f_1} and n_{f_2} represent the number of filters in the first and second convolutional layers, respectively. The configurations are as follows: TM-CNN($n_{f_1} = 8$, $n_{f_2} = 16$), TM-CNN($n_{f_1} = 16$, $n_{f_2} = 32$), and TM-CNN($n_{f_1} = 32$, $n_{f_2} = 64$).

The results suggest that the TM-CNN($n_{f_1} = 16$, $n_{f_2} = 32$) configuration achieves the highest precision, recall, and F1 score, with values of 0.8526, 0.8657, and 0.8591, respectively. The other two configurations, although comparable, exhibit slightly lower performance metrics.

These findings contribute to the model selection process by identifying the filter configuration of the TM-CNN that leads to the best performance when paired with the EAGNN$_5$ backbone, under the given node and edge feature settings.

	Precision	**Recall**	**F1**
TM-CNN($n_{f_1} = 8$, $n_{f_2} = 16$)	0.8469	0.8558	0.8513
TM-CNN($n_{f_1} = 16$, $n_{f_2} = 32$)	**0.8526**	**0.8657**	**0.8591**
TM-CNN($n_{f_1} = 32$, $n_{f_2} = 64$)	0.8394	0.8606	0.8499

Table A.1: Performance metrics for different filter configurations of the TM-CNN model in combination with the EAGNN$_5$ backbone. NF=N_1 and EF=$E1234$. Results are shown for the test set.

A.2. Ablation study

The Ablation Study section presents the examination of the effects of different node feature (NF) and edge feature (EF) configurations on model performance, primarily through two tables.

Table A.2 depicts an ablation study on the impact of various node feature configurations on the performance of the test set. Five different settings are explored using EAGNN_2 and EAGNN_5 models in combination with a TM-CNN classifier. These configurations involve different combinations of node features, designated as N_1, N_{14}, N_{12}, and N_{123}, with N_1 representing Node types. The edge features remain constant at E_{1234} for all configurations in this study. According to the results, the best performance across all metrics—precision, recall, F1 score, and accuracy—is achieved with the EAGNN_5 model when employing N_1 node features. It's noteworthy that using N_4, which signifies DL extracted features, resulted in lower performances.

Model	Classifier	NF	EF	Precision	Recall	F1	ROC-AUC	Accuracy
EAGNN_2	TM-CNN	N_1	E_{1234}	0.8523	0.8595	0.8559	0.9866	0.9837
EAGNN_2	TM-CNN	N_{14}	E_{1234}	0.8152	0.8006	0.8078	0.9731	0.9785
EAGNN_5	TM-CNN	N_1	E_{1234}	**0.8526**	**0.8657**	**0.8591**	0.9874	**0.9840**
EAGNN_5	TM-CNN	N_{12}	E_{1234}	0.8202	0.6224	0.7077	0.9765	0.9710
EAGNN_5	TM-CNN	N_{123}	E_{1234}	0.8476	0.8566	0.8520	**0.9879**	0.9832

Table A.2: Ablation study examining the impact of different node feature (NF) configurations on the test set performance.

Table A.3 carries out a similar ablation study, but this time focusing on the influence of different edge feature configurations on the test set performance. All experiments in this study use the EAGNN_5 model and a TM-CNN classifier, along with the N_1 node feature. The configurations differ in their edge features, marked as E_{1234}, E_{123}, E_{12}, and E_1. According to the results, the model performs best when the E_{1234} edge feature configuration is used, demonstrating the highest precision, recall, F1 score, and accuracy. It's important to mention that the E_5 configuration, which would represent aggregating edge features with a $p = 512$ channels (where p denotes the dimension of the features extracted using the pretrained ResNet18 model), has not been included in the table due to the significant computational cost of such a procedure, resulting in a out of memory error when aggregating information for updating node embeddings.

These ablation study tables offer valuable insights into how different configurations of node and edge features can influence model performance. They help in identifying the most effective combinations of features for this particular classifier and the dataset under

Model	Classifier	NF	EF	Precision	Recall	F1	ROC-AUC	Accuracy
$EAGNN_5$	TM-CNN	N_1	E_{1234}	**0.8526**	**0.8657**	**0.8591**	**0.9874**	**0.9840**
$EAGNN_5$	TM-CNN	N_1	E_{123}	0.8360	0.8525	0.8442	0.9859	0.9823
$EAGNN_5$	TM-CNN	N_1	E_{12}	0.8338	0.8390	0.8364	0.9852	0.9815
$EAGNN_5$	TM-CNN	N_1	E_1	0.8093	0.8386	0.8237	0.9821	0.9798

Table A.3: Ablation study assessing the effect of different edge feature (EF) configurations on the test set performance.

study.

As informed by [27], they confirm that the most effective features combination for this specific task and the tested dataset is the N_1 node feature paired with the E_{1234} edge feature configuration.

B | Appendix B

B.1. Graph Segmentation

To enhance comprehension of the model predictions on the graph, a segmentation method is introduced. Considering that the graph generated by Delaunay triangulation consists of Triangular Subgraphs, which correspond to combinations of 3 edges in a graph, these triangles are then color-coded to represent healthy or degraded basement membranes (BMs).

Triangles are segmented as follows:

(Empty) Triangles with 0 BM crossing edges

(Red) Triangles with only 1 BM crossing edge

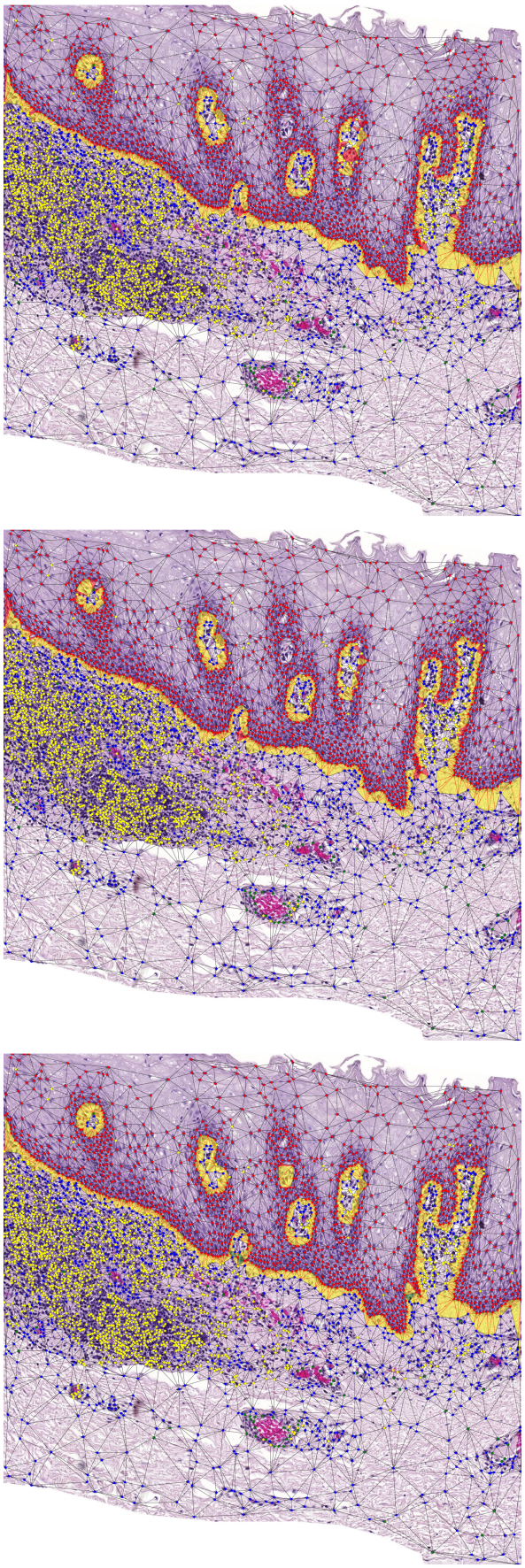
(Yellow) Triangles with 2 BM crossing edges

(Green) Triangles with 3 BM crossing edges

Notably, red triangles serve as indicators of compromised BMs. In these cases, the BM enters the triangle via one edge, but does not exit, indicating an interruption or break (with the exception of rare cases where the BM enters and exits through the same edge, which can be filtered out). Conversely, yellow and green triangles are indicative of contiguous BM structures, with the BM entering the triangle through one edge and exiting via another. Specifically, green triangles are cases where the BM enters through one edge, exits, re-enters through a different edge, and exits again through the third edge.

Figures B.1, B.2, and B.3 provide instances of both healthy and damaged basement membranes (BMs) extracted from the oral tissue samples of three distinct patients. For each sample, the presentations are categorized into the ground truth segmented graph, the segmented graph inferred from the predictions of the model proposed by [27], and the segmented graph deduced from the predictions of the model presented in this study. It's crucial to note the manner in which the ground truth segmentation helps elucidate the nature of graph segmentation. A continuous path delineated within yellow triangles illus-

trates an intact BM localized in that segment of the graph. Conversely, the presence of a red triangle indicates a deterioration of the BM leading to a break, thereby disrupting the "triangle path".



(a) Ground Truth (b) (EAGNN$_2$,BC+MLP)[27] (c) (EAGNN$_5$,TM-CNN)

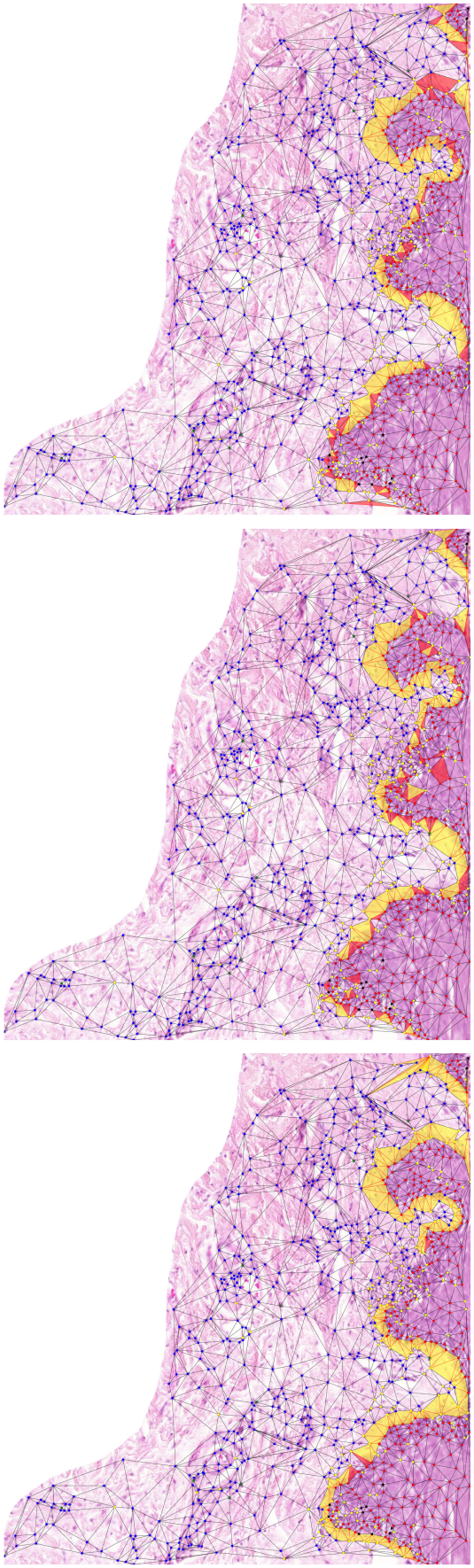
Figure B.1: Segmentation of an healthy Oral Tissue's graph

(a) Ground Truth

(b) (EAGNN$_2$,BC+MLP)[27]

(c) (EAGNN$_5$,TM-CNN)

Figure B.2: Segmentation of an Oral Tissue's graph presenting BM breakages



(a) Ground Truth (b) (EAGNN$_2$,BC+MLP)[27] (c) (EAGNN$_5$,TM-CNN)

Figure B.3: Segmentation of an Oral Tissue's graph presenting BM breakages

List of Figures

2.1 Two examples of H&E staining of oral mucosa samples 7
2.2 Whole slide images (WSIs) of a healthy oral mucosa (top) and inflammed oral mucosa (bottom). The boxes show the magnified areas. Arrows point to key cell types, epithelial tissue (green), fibroblasts (yellow), endothelial cells (blue) and inflammation (red). The finger structure of the lamina propria can be seen as well as the basal layer of the epithelial. The very densely packed cells of the basal layer can be seen in the magnified images and how they diverge from each other when approaching the top. 8
2.3 A basic Perceptron model. 13
2.4 A Multi-layer Perceptron (MLP) with one hidden layer. 14
2.5 Illustration of a neural network with and without Dropout. 17
2.6 Simplified architecture of a Convolutional Neural Network (CNN). 19
2.7 An illustration of the convolution operation. A 3x3 filter K slides over the input image I, and for each position, an element-wise multiplication followed by a summation (dot product) is performed to produce the feature map $I * K$. 20
2.8 An illustration of the pooling operation with 2×2 filter and stride 2. (Top-Right) Result of Max-pooling. (Bottom-Right) Result of Mean-Pooling. . . 21
2.9 *(Left)* Example of undirected graph with four nodes and four edges. *(Right)* Adjacency matrix of the graph. 21
2.10 Two-dimensional convolution vs graph convolution [75] 24
2.11 A representation of different graph-based models for histopathology images. A) Cell-graph representation for prostate cancer. B) Tissue-graph representation for colorectal cancer. C) Hierarchical cell-to-tissue graph representation for breast cancer. (Source: [3]) 27

3.1 (**a**) WSI of healthy buccal oral mucosa that has been segmented into tiles of 2000 × 2000 pixels. (**b**) Annotation of the extension of the BM visualised with blue line. (**c**) Nuclei centroids were annotated and labelled as either epithelial (red), fibroblast or endothelial (blue), inflammatory (green) or lymphocytic (yellow) (Image taken from [48]) 37
3.2 Cell-graph model generation. (Left) Manually Annotated Tile. (Right) Generated Graph; (red) BM crossing edges, (black) BM non-crossing edges. 39
3.3 EAGNN architecture overview. 42
3.4 Triangular Motifs CNN (TM-CNN) architecture overview. 43

4.1 Comparison of ($EAGNN_2$,BC+MLP) [27] and ($EAGNN_5$,TM-CNN) predictions for healthy BM tissue sample. The blue line is the BM annotation. The green, yellow and red lines represent true positive, false negative and false positive edge predictions respectively 56
4.2 Comparison of ($EAGNN_2$,BC+MLP) [27] and ($EAGNN_5$,TM-CNN) predictions for degraded BM tissue sample. The blue line is the BM annotation. The green, yellow and red lines represent true positive, false negative and false positive edge predictions respectively 57
4.3 Segmentation of an healthy Oral Tissue's graph. Red triangles indicate disruptions in the BM structure, while yellow and green triangles indicate BM continuity. 58
4.4 Segmentation of an Oral Tissue's graph presenting BM breakages. Red triangles indicate disruptions in the BM structure, while yellow and green triangles indicate BM continuity. 59

B.1 Segmentation of an healthy Oral Tissue's graph 85
B.2 Segmentation of an Oral Tissue's graph presenting BM breakages 86
B.3 Segmentation of an Oral Tissue's graph presenting BM breakages 87

List of Tables

2.1 Histological grading criteria for defining features of oral mucosal cGVHD [68] 11

3.1 Distribution of edge classes across different splits 38

4.1 Model Configuration. (EAGNN$_i$) NF and EF represent respectively node and edge features considered in the study, and their dimension determined respectively the Node and the Edge Feature Matrices input dimensions. . 50

4.2 Performance metrics of the various (EAGNN$_i$,TM-CNN) configurations on the validation set. The metrics are employed to ascertain the optimal number of aggregation layers (i) within the EAGNN backbone model. Node Features (NF) and Edge Features (EF) represent aggregated node and edge characteristics, respectively. 52

4.3 Precision, Recall and F1 score metrics of the (EAGNN$_5$,TM-CNN) configuration with varying threshold τ on the validation set. The metrics are used to determine the optimal value for the threshold τ. 52

4.4 Benchmarking of GNN models, edge classifiers, and aggregated node and edge features for BM prediction on the oral mucosa cell-graph dataset. . . 53

A.1 Performance metrics for different filter configurations of the TM-CNN model in combination with the EAGNN$_5$ backbone. NF=N_1 and EF=$E1234$. Results are shown for the test set. 79

A.2 Ablation study examining the impact of different node feature (NF) configurations on the test set performance. 80

A.3 Ablation study assessing the effect of different edge feature (EF) configurations on the test set performance. 81

Acknowledgements

First and foremost, I would like to extend my deepest gratitude to my advisor, Prof. Daniele Loiacono. His guidance, particularly during the thesis writing process, has been invaluable.

Special thanks are due to my co-advisors, Karl Meinke of the KTH division and Rachael Sugars of the KI division. Karl has been instrumental in my technical development, encouraging me to think outside the box and delve deep into the intricacies of my research model. His feedback and unique perspectives on my work have been enlightening. Rachael provided me with a foundational understanding of the life sciences and medical frameworks relevant to my project. Her feedback has been consistently constructive, and her expertise significantly enriched the scope of this research.

I would also like to acknowledge the support and contributions of my research team colleagues: Aravind Nair (KTH), Helena Arvidsson (KI), and Nikolce Tudzarovski (KI). Their collective wisdom and collaborative efforts have been pivotal in shaping and advancing my project.

I wish to express my gratitude to KTH Royal Institute of Technology and Digital Futures for the research funding that made this work possible.

On a personal note, my heartfelt thanks go to my family—my mum, dad, and sister—for their unwavering love and support throughout my educational journey. Your encouragement has been my backbone, and I am forever grateful for your belief in me.

To my friends back in my hometown, thank you for always being there, no matter the distance. Special thanks go to my BJJ friends in Milan from L'Accademia, who helped me find my footing during my first year in Milan.

Lastly, but certainly not least, my gratitude extends to all the new friends I made during my year abroad in Stockholm. A special mention must be made for Prana Jiu-Jitsu; you became my second home in Stockholm and supported me both on and off the mat.