Utrecht University

Department of Information and Computing Science

Applied Data Science Master Thesis

Facilitating Neuro-Oncology Research: An Extensible Graph Neural Network Framework for Brain Tumour Classification

First Examiner:

Artem Kaznatcheev

Second Examiner:

Sanna Abeln

Candidate:

Alejandro Alarcón Torres

In cooperation with:

Sebastian Waszak, NCMM

Birgit Kriener, NCMM

July 12, 2023

Acknowledgements

First and foremost, I would like to express my gratitude to my advisor, Artem Kaznatcheev, for his support, encouragement, and guidance throughout the course of this research.

I also wish to extend my sincere appreciation to Sebastian Waszak and Birgit Kriener, who, in collaboration with the Norway Centre for Molecular Medicine (NCMM), have made invaluable contributions to my project. Their insights, advice, and collaborative spirit have played a pivotal role in shaping this research and have enriched my academic experience tremendously.

Furthermore, I would like to thank the Norway Centre for Molecular Medicine (NCMM) for its support and collaboration. The Centre's commitment to fostering research excellence and its focus on advancing the frontiers of molecular medicine have been instrumental to the progression and success of my project.

Abstract

This study addresses the necessity for precise and efficient brain tumour classification techniques, traditionally dependent on manual histopathology, a process prone to inaccuracies due to subtle differences in images.

Edge-definition techniques in cell-graphs play a fundamental role in graph-based learning, as they encode the interaction between the tumours' cells that can be crucial for capturing complex histopathological patterns. These patterns, when accurately identified, can provide valuable insights into tumour structure and potential malignancy, therefore enhancing the precision of cancer diagnosis and prognosis.

The potential of Graph Neural Networks (GNNs) is further explored within this context. Recognizing the diversity and complexity of brain tumours, we leveraged the flexibility and extensibility of the GraphGym framework in our method, which allowed for a more comprehensive and nuanced approach to brain tumour classification. The resultant framework is used to evaluate both the performance of edge-definition approaches and the effectiveness of different GNN architectures.

The objective is to identify the most effective combination for brain tumour classification. The results of this study aim to provide significant insights and make a substantial contribution to the advancement of diagnostic accuracy in neuro-oncology.

Table of contents

1 Introduction

Brain tumours, a category of cancers affecting millions globally, present a significant concern in contemporary healthcare. Their high degree of variability, the severity of symptoms, and the profound impact on quality of life make them a critical area of focus in ongoing oncological research. The complexity of diagnosis and treatment further underscores the need for innovative approaches that enhance accuracy and efficiency.

Understanding the biomedicine involved in brain tumours is crucial to our study. Brain tumours arise from several types of cells within the brain and can be broadly categorized as primary or secondary, benign, or malignant, each with distinct tissue characteristics (Doolittle, 2004). Histopathology (Section 2.1), which involves the examination of tissue samples under a microscope, plays a vital role in the definitive diagnosis and characterization; it is considered the gold standard for tumour diagnosis and classification (Charles et al., 2011). By analysing tissue samples, pathologists can assess the cellular composition, architectural patterns, and molecular features of brain tumours. Histopathological analysis allows for the identification of specific tumour types, such as gliomas, which will be the target of this project (Section 1.1).

While histopathology has been instrumental in diagnosing brain tumours, it comes with certain challenges. The images that are dealt with in histopathology are vast and rich in detail, which, while providing comprehensive data, also makes them computationally heavy and challenging to process efficiently (Shetty et al., 2022). Furthermore, subtle differences between tumour types can be difficult to discern and quantify, leading to potential discrepancies in diagnosis (Shetty et al., 2022).

By transforming these high-resolution images into graph structures, we can capture the key features of the tumour while reducing the computational overhead associated with processing such images (Zheng et al., 2022). Graph Neural Networks (GNNs) offer a powerful framework to analyse graph-structured data, enabling the extraction of meaningful representations and accurate predictions (Wu et al., 2021). This graph-based approach allows us to exploit the complex relationships and dependencies within the tumour microenvironment, facilitating comprehensive brain tumour classification and providing valuable insights for diagnosis and treatment planning (Doi, 2007).

A major component of this process involves the edge-definition in cell-graphs, which models the complex interactions between tumour cells. The accurate definition of these edges is critical to capturing the intricate cellular interactions and dependencies. However, this task poses a significant challenge due to the inherent complexity and variability of tumour tissues (Nath, 2022). Determining the most effective approach for edge-definition is a central aspect of our study.

GNNs are a subclass of neural networks designed specifically to work with graph data. They operate by passing messages between nodes in a graph, enabling them to capture and learn from the complex patterns and relationships within the data (Sanchez-Lengeling et al., 2021). While GNNs offer several advantages, such as their ability to handle irregular data and their flexibility in terms of input size (Sanchez-Lengeling et al., 2021), they also present certain challenges, including a higher degree of complexity and potential difficulties in model interpretation (Ahmedt-Aristizabal et al., 2022).

The aim of this project is to provide a framework that enables the exploration of the viability and effectiveness of GNNs for brain tumour classification. To develop our approach, we have adapted and extended the GraphGym framework (You et al., 2020), a flexible and extensible platform for graph-based learning.

The resulting framework will be accessible to the NCMM and will encompass all the essential steps from graph generation to batch training and testing of GNN models with a wide range of easily customizable parameters. Furthermore, it aims to establish a standard method for evaluating and comparing both edge-definition techniques and GNN architectures. We also present an evaluation of some of the most promising GNN architectures under varying settings.

1.1 Neuro-oncological context

Brain cancer encompasses a broad range of neoplastic growths that develop within the brain. These abnormal cell growths can arise from the brain itself (primary brain tumours) or spread from other parts of the body (secondary or metastatic brain tumours) (Al-Hussaini, 2013). Brain tumours pose significant challenges due to their intricate location within the central nervous system and the potential for invasive behaviour. Invasive cells can migrate away from the primary tumour site and invade healthy brain tissue, making it difficult to remove all the tumour during surgery (Cheng et al., 2011).

Brain tumours can be classified into several distinct types based on various factors, including their origin, histological features, and genetic characteristics (Al-Hussaini, 2013). Among these, gliomas are the most prevalent type of primary brain tumour and arise from glial cells[1] (Mesfin & Al-Dhahir, 2023). Their versatile roles in brain function underline the complexity of gliomas that originate from these cells.

[1] Glial cells aid in maintaining homeostasis, providing structural support, and regulating the speed of electrical impulses traveling along neurons by forming the myelin sheath (Adrienne Dellwo, 2023). Furthermore, glial cells are responsible for cleaning up cellular debris and pathogens, providing nutrients to neurons, and participating in brain immune responses.

Gliomas can be categorized by their grade. In the context of brain tumours, grade refers to the degree of malignancy of the tumour and is determined by examining the tumour cells under a microscope and assessing how abnormal they appear compared to normal cells. A key aspect of this assessment is the observation of mitotic activity, which is the process of cell division. High mitotic activity indicates that cells are dividing rapidly, a characteristic of aggressive tumours. The higher the grade, the more aggressive the tumour is likely to be, and the more rapidly it is likely to grow and spread (*Brain Tumours - NHS*, 2023).

High-Grade Gliomas: High-grade gliomas are more aggressive and grow faster. Cells are undifferentiated and highly malignant, thus more likely to spread to other parts of the brain or spinal cord, having a worse prognosis.

Low-Grade Gliomas: Low-grade gliomas are less aggressive and grow more slowly. They are less likely to spread to other parts of the brain or spinal cord (David Reardon, 2019). In a Low-Grade Glioma, cells are well-differentiated and exhibit less aggressive tendencies, having a better prognosis.

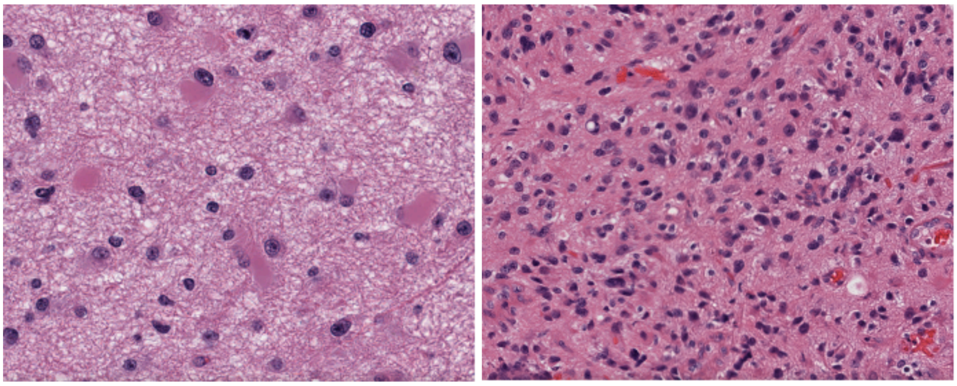


Figure 1. Histopathological images of Gliomas. Left: Low-Grade Glioma. Right: High-Grade Glioma.

1.2 Graph Based Learning

Graph-based learning, an emerging paradigm in machine learning, leverages the underlying relationships and interactions between data points to generate comprehensive, nuanced insights. Predominantly utilized in the realm of deep learning, it's a method that adopts the graph theory concept to create models and algorithms that not only account for individual data entities, but also for the interconnections amongst them.

Graph Neural Networks (GNNs), a subset of graph-based learning, have arisen as a powerful tool to handle data structured as graphs. Data in the form of graphs encapsulates relationships between entities (depicted as nodes) using edges. This type of data representation is incredibly versatile, permitting the modelling of a myriad of complex systems (Wu et al., 2021).

Unlike traditional neural networks that require a fixed-size input, GNNs can process graphs of any size and shape, ensuring its adaptability across varying scopes of data (Sanchez-Lengeling et al., 2021). They preserve the relational information between nodes and allow the propagation of information along the edges, thereby capturing the global and local dependencies of the graph structure. This feature is instrumental in understanding the relationship between different regions of interest in complex structures.

In the medical imaging domain, these networks have demonstrated exceptional promise (Levy et al., 2021; Sudbø et al., 2000; Wang et al., 2022). By preserving spatial and structural context, GNNs can potentially increase the accuracy of classification tasks, offering significant improvements over traditional machine learning techniques (Chan et al., n.d.).

In Figure 2, we can observe the variations in distances, shapes, and sizes across the nodes and edges, which may serve as valuable features for the graph-based model. These characteristics can vary considerably and may carry specific information that could be characteristic of tumour subtypes.

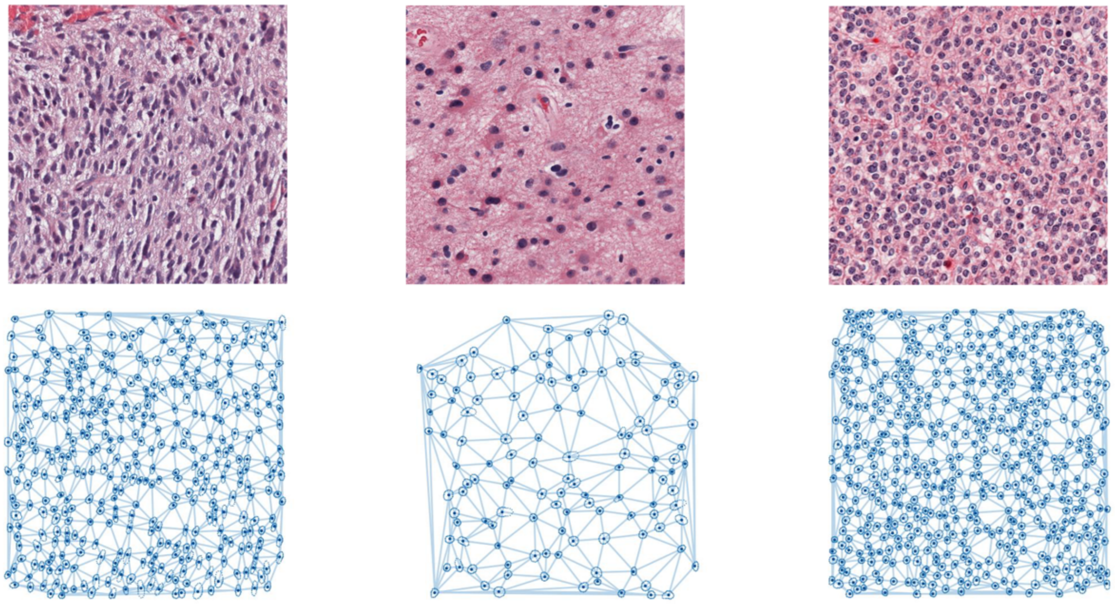


Figure 2. TOP: Examples of WSIs of high-grade glioma (left) and low-grade gliomas (middle and right). BOTTOM: Graphs induced from the WSIs of the upper row using Delaunay Triangulation. Courtesy of Birgit Kriener, NCMM.

Preliminary explorations using slide-wide bulk features (e.g., mean and variance of the distance distribution and node density, mean and variance of Hu-moments) have hinted at possible clustering of tumour subtypes, but the results were not distinct enough to rely on these features alone.

Note the challenge of classifying the graphs into high- and low-grade categories based on simple graph metrics such as node and edge densities. GNNs, with their ability to capture more nuanced patterns and relationships in the data, are expected to perform better in grasping these informative subtleties than broad averages on a slide-wide basis.

This study aims to harness the potential of GNNs to improve brain tumour classification, an application of profound significance in medical imaging and oncology. By exploring the synergistic relationship between the versatile nature of GNNs and the intricate topology of brain tumours, this study takes a promising step in exploring new methodologies for brain cancer classification.

2 Background

2.1 Histopathology

Histopathology is a branch of pathology that involves the microscopic examination of tissues to study and diagnose diseases. It combines the fields of histology, which focuses on the microscopic structure of tissues, and pathology, which deals with the nature and causes of diseases.

In the context of brain tumour analysis, histopathology plays a crucial role in understanding the characteristics and behaviour of tumours (Al-Hussaini, 2013). When a brain tumour is suspected, a biopsy or surgical resection is performed to obtain a tissue sample. The obtained samples are then processed and prepared for histopathological analysis.

During the preparation, the tissue sample is physically processed and sliced into thin sections, which are then stained to enhance specific cellular components. These stained tissue sections are mounted on glass slides and examined under a microscope in order to determine the characteristics of the tumour.

Histopathological analysis has traditionally been a manual procedure that relies on the expertise of pathologists. However, with the advent of Whole Slide Imaging (WSI), entire glass slides containing tissue samples can be scanned and converted into high-resolution images (see Figure 3), allowing for digital processing and analysis, also called digital pathology.

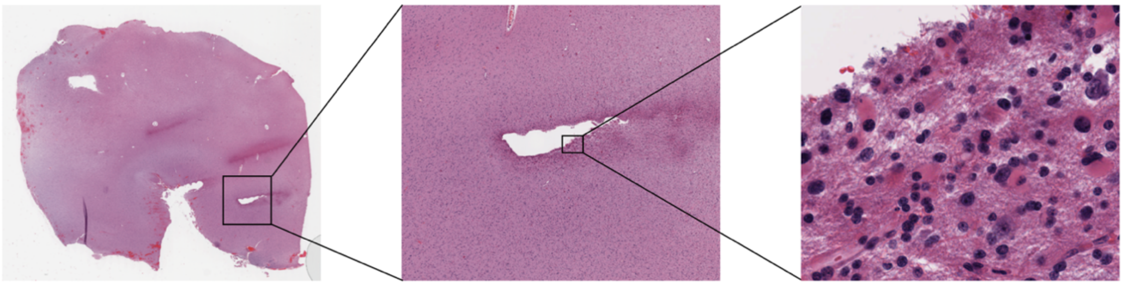


Figure 3. Example of a Whole Slide Image of a Glioblastoma at increasing augmentations (left to right). At highest augmentation we can observe the tumour's nuclei in dark purple. Note that other cellular components are hardly distinguishable, making it hard to delineate cellular boundaries.

2.2 Graph Neural Networks

Graph Neural Networks (GNNs) are a class of deep learning models specifically designed to handle graph-structured data. GNNs have gained significant attention in recent years for their ability to capture the complex relationships and dependencies present in graph data (Mohi ud din & Qureshi, 2022).

GNNs extend traditional neural network architectures by incorporating message passing and aggregation mechanisms that allow nodes in the graph to exchange information with their neighbouring nodes (Sanchez-Lengeling et al., 2021). By iteratively propagating and updating information across the graph, GNNs learn to capture both local and global dependencies, enabling accurate predictions and analysis.

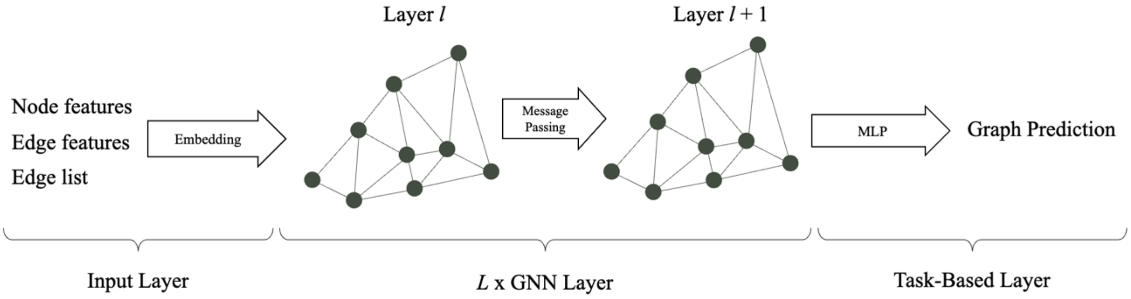


Figure 4. Overview of the general structure of Graph Neural Networks, which embed the graph with the input layers, perform neighbourhood diffusion in the GNN layers, and finally make a prediction through a task-based MLP layer.

GNNs embrace a whole bunch of architectures that can be told apart from each other by the mechanism that is used for message passing. The following sections will try to dive deeper into the general structure of GNNs, shown in Figure 4, while also giving more details on the message passing layers, which are at the core of the GNN models, and determine the characteristic behaviour of the different GNN architectures.

2.2.1 Input Layers

In GNN models, the graphs are represented as a collection of node features, an edge list specifying the source and target of all edges, and optionally, an edge feature list providing detailed information about each edge. However, this raw data structure is not directly suitable for the Message Passing layers of the GNN framework.

To overcome this, we employ the concept of 'embeddings' facilitated by input layers. An embedding in this context is a transformed, dense vector representation of the node or edge that can capture the essential characteristics of the original graph elements. These embeddings are more amenable to machine learning algorithms and are an efficient way to represent the complex data in a format that the model can process.

So, the role of the input layers is to act as an interface that translates this raw graph data into a form the model can work with. These layers effectively encode the graph into a set of initial embeddings, assigning each node and edge a feature representation, or 'embedding', that serves as the starting point for the complex computations that take place in the subsequent layers of the GNN model.

2.2.2 Inside the core of GNNs

At the core of the GNNs models, we find a neighbourhood aggregation mechanism that is encoded in the message passing layers and is responsible for aggregating feature information from a node's immediate neighbours.

Each message passing layer computes a representation for the nodes and edges of the graph through recursive neighbourhood diffusion, where each node creates a new feature representation that encapsulates not only its own features but also the features of its neighbouring nodes. Stacking L layers allows the network to build node representations from the L-hops neighbourhood of each node. This way, nodes can "communicate" with each other through this feature exchange process, which enables GNNs to consider the overall topology of the graph in their computations.

At a high level, the message passing mechanism involves two main steps: message construction and message aggregation. In the message construction step, every node computes a message for its neighbouring nodes. Next, in the message aggregation step, each node aggregates the messages it receives from its neighbours, as shown in Figure 5. The aggregation functions must be invariant to both the number of neighbours and its ordering. The specific way in which messages are generated and aggregated can vary and depend on the particular architecture of the GNN.

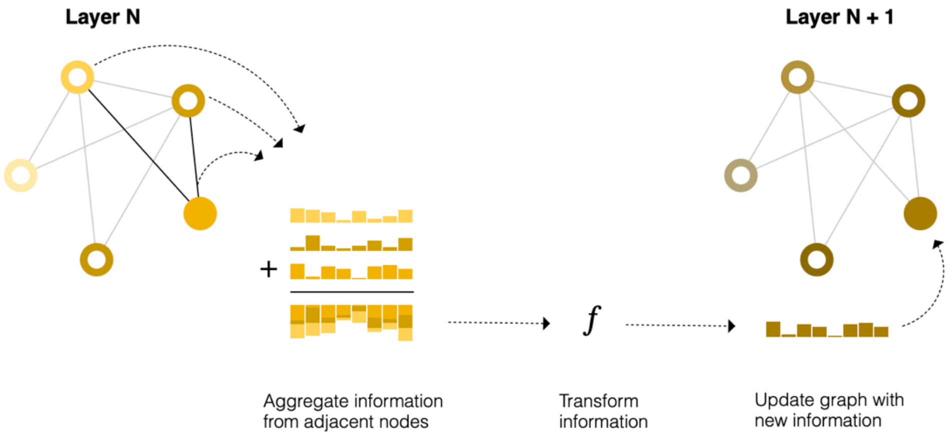

Layer N
Layer N + 1
+
f
Aggregate information from adjacent nodes
Transform information
Update graph with new information

Figure 5. Illustration of the message massing mechanism between two layers. Extracted from (Sanchez-Lengeling et al., 2021)

In the following subsections, we describe in more detail the message passing layers that we will benchmark in this project, and that have been selected according to the classification task that was introduced by NCMM. Each of these layers employs different aggregators to acquire information from each node's neighbours and have been selected based on recent survey studies where they show promising results in other oncological areas (Ahmedt-Aristizabal et al., 2022).

2.2.2.1 Graph Convolutional Networks

Graph Convolutional Networks (GCNs) (Kipf & Welling, 2016) are a type of GNN that effectively capture the complex patterns and dependencies within graph data by leveraging the principles of convolution from traditional Convolutional Neural Networks (CNNs) and adapting them to work with graph structures.

GCNs operate on an input graph where each node is associated with a feature vector. The key operation of a GCN layer is to aggregate the features of neighbouring nodes to update the feature of each node through the message passing mechanism.

2.2.2.2 Graph Sample and Aggregation

Graph Sample and Aggregation (GraphSAGE) (Hamilton et al., 2017) is another variant of GNNs that has demonstrated success in learning on graph-structured data. Unlike many other GNN architectures such as GCNs, which are designed to work with a fixed graph structure, GraphSAGE is designed to generate embeddings for nodes in unseen data, enabling inductive learning on large graphs.

The fundamental operation of GraphSAGE involves learning a function that generates a node's embedding by sampling and aggregating features from its local neighbourhood. Specifically, the GraphSAGE algorithm updates the feature vector of each node based on its own features and the aggregated features of its neighbours.

The updating of node features in a GraphSAGE layer can be described by the following steps:

Neighbourhood Sampling: For each node, a fixed-size neighbourhoods is sampled. The size of the neighbourhoods is a hyperparameter that determines the number of neighbours to consider.

Aggregation: The features of the sampled neighbours are aggregated using an aggregation function, such as mean, max, or LSTM. The aggregation operation captures the local structural information around each node.

Combination: The aggregated neighbourhood features are then combined with the features of the node itself. This is typically done through concatenation followed by a linear transformation and a non-linear activation function.

Normalization: Optionally, the resulting node embeddings can be normalized (e.g., using L2[2] normalization) to prevent the scale of the embeddings from growing with the depth of the network.

2.2.2.3 Graph Attention Networks

Graph Attention Networks (GAT) (Veličković et al., 2017) are a novel variant of GNNs that leverage the power of attention mechanisms, a concept that has proven highly successful in various domains such as natural language processing. The key innovation in GAT is that it introduces the attention mechanism into GNNs, allowing for a more dynamic and adaptive aggregation of neighbourhood information.

The primary operation of a GAT layer involves computing a set of attention coefficients for each node and its neighbours. These coefficients determine the importance of each neighbour’s features when updating the node's features. The attention mechanism allows the model to focus more on the important neighbours and less on the less important ones, providing a more nuanced non-isotropic aggregation of neighbourhood information compared to methods that treat all neighbours equally.

The computation of the attention coefficients in a GAT layer can be described by the following steps:

Attention Score Calculation: An attention score is calculated for each edge (i.e., pair of nodes), indicating the importance of the connection. This score is usually calculated using a shared attentional mechanism, such as a single-layer feedforward neural network with a LeakyReLU[3] activation function.

[2] L2 normalization, also known as least squares or Euclidean normalization, is a technique used in machine learning and data preprocessing to scale the values of a vector so that the sum of the squares of its elements equals 1 (Yang et al., 2020).

[3] Leaky ReLU (Rectified Linear Unit) (J. Xu et al., 2020) is an improved version of the ReLU activation function. The primary purpose of introducing Leaky ReLU is to address the "dying ReLU" problem, which occurs when the learning rate is too high or there is a large negative bias, causing some neurons to become inactive and produce zero gradients.

Normalization: The attention scores for each node and its neighbours are normalized using the softmax function, resulting in the attention coefficients.

Feature Aggregation: The features of each neighbour are multiplied by the corresponding attention coefficient and then summed, resulting in the new feature vector for each node.

2.2.2.4 Graph Isomorphism Networks

Graph Isomorphism Networks (GIN) (Xu et al., 2018) are a powerful class of GNNs that have been designed to capture the expressive power of the Weisfeiler-Lehman (WL) test[4] of isomorphism on graphs. GINs are particularly interesting because they have been theoretically shown to be as powerful as the WL test in distinguishing different graph structures.

The core operation of a GIN layer involves aggregating the features of each node and its neighbours, and then updating the node's features based on this aggregated information. Unlike some other GNN models, GIN introduces a learnable parameter that allows it to control the importance of a node's own features versus the features of its neighbours when updating the node's features, providing a more flexible and adaptive aggregation process.

2.2.2.5 Normalization and Residual Corrections

As a final note on the GNN component, each message passing layer can be augmented with batch normalisation (BN) (Ioffe & Szegedy, 2015) and residual connections (He et al., 2015) in order to improve both the performance as well as the training of the models.

Batch Normalization

Batch normalization is a technique used to make the training of artificial neural networks faster and more stable through normalization of the layers' inputs by re-centring and re-scaling. It was proposed to address the issues related to the training of deep neural networks, such as internal covariate shift,

[4] The Weisfeiler-Lehman (WL) (Douglas, 2011) test is a heuristic method for graph isomorphism testing, which aims to determine if two graphs are isomorphic or not (Zemlyachenko et al., 1985). Graph isomorphism is the problem of determining whether there exists a bijection between the vertex sets of two graphs that preserves the adjacency relations. The interpretation of the WL test is as follows:

- If the test returns false, then the two graphs are surely not isomorphic.
- If the test returns true, then the two graphs may be isomorphic.

where parameter initialization and changes in the distribution of the inputs of each layer affect the learning rate of the networks (Ioffe & Szegedy, 2015).

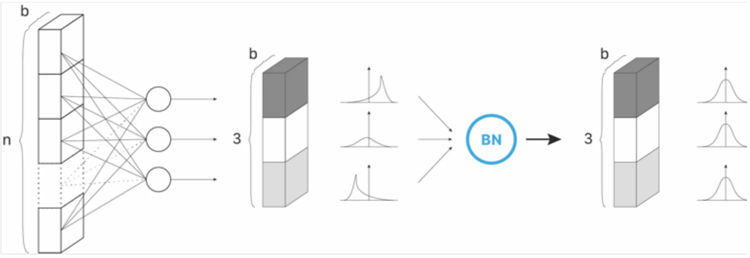


Figure 6. Example of a 3-neurons hidden layer, with a batch of size b. Each neuron follows a standard normal distribution. Extracted from Johann Huber, 2020.

Batch normalization offers several benefits for GNNs:

Faster training. Although each training iteration might be slower due to the extra calculations during the forward pass and the additional hyperparameters to train during backpropagation, the overall training process converges more quickly.

Higher learning rates. Gradient descent usually requires small learning rates for the network to converge. As networks get deeper, their gradients get smaller during backpropagation, requiring even more iterations. Batch normalization allows the use of higher learning rates, further increasing the speed at which networks train.

Easier weight initialization. Weight initialization can be difficult, especially when creating deeper networks. Batch normalization helps make weight initialization less critical.

Residual Connections

In the GNN framework, residual connections are a technique used to improve the performance and training of deep neural networks. They are a type of skip connection that allows the network to learn residual functions with reference to the layer inputs, instead of learning unreferenced functions. As shown in Figure 7, Residual connections provide an alternative path for data to reach latter parts of the neural network by skipping some layers.

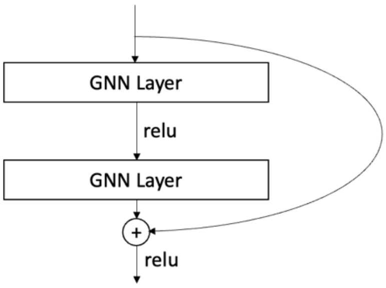


Figure 7. Example of a Residual Connection.

This helps in mitigating the vanishing gradient problem, which occurs when gradients become too small as they backpropagate through the network, making it difficult to train deep models. By allowing gradients to flow directly through the network without passing through non-linear activation functions, residual connections help in training deeper models more effectively.

2.2.3 Task-Based Layers

The final component in the structure of GNNs is a Multi-Layer Perceptron (MLP) that is in charge of computing task-dependent outputs. This component takes the final representation or embedding that results from the last message passing layer and is trained to minimize the error of the task that the model is trained for.

2.3 Entity Representation

Whole Slide Imaging (WSI) is a powerful technique that digitalizes histopathology slides at a high resolution, yielding a vast and detailed dataset that contains critical information about tissue and cell structures. However, transforming these highly dimensional and complex datasets into a graph representation for further processing by GNNs can pose challenges (Pantanowitz et al., 2015). Among the many challenges, defining the entities (nodes) and the relationships between them (edges) within a graph is a fundamental one (Chan et al., n.d.).

Node definition usually involves representing each cell as a node. In the case of WSI, the segmentation step, where cells are identified and separated from the surrounding tissue, often determines node creation. Each node is then assigned features that describe the cell's morphological and phenotypical properties such as size, shape, or texture.

Edge definition, on the other hand, is a more intricate problem, as it involves defining a relationship between nodes. Edge definition should be a deliberate process since it models the critical interactions between cells within the tumour environment (Rathinabai & Jeyakumar, 2021). This interaction could be determined based on a variety of factors, such as spatial proximity, phenotypical similarity, or specific biological interactions.

Some of the state-of-the-art approaches for defining edges that model cell interaction are:

Voronoi Tessellation: A Voronoi diagram (Figure 8) (Aurenhammer, 1991) is a partition of a plane into regions based on the distance to a given set of seeds. Each region consists of all points closer to a specific seed than to any other seed. Voronoi diagrams can be used to derive graphs from WSIs, where each seed represents a nucleus in our tissue, and the edges are determined by the adjacency of the regions. Voronoi diagrams can be particularly useful for outlining cell borders in histologic sections, which can be challenging to recognize otherwise (Bock et al., 2009).

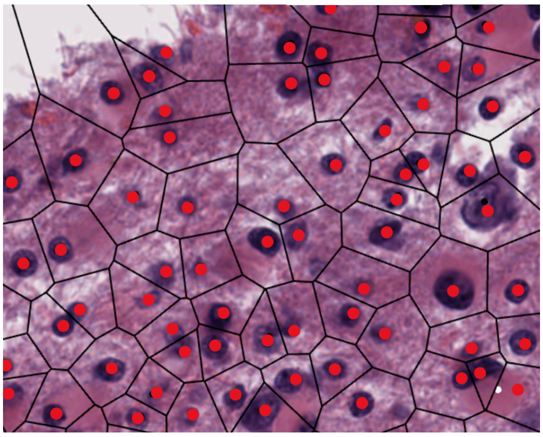


Figure 8. Voronoi tessellation superposed on the original WSI using the nuclei as seeds for the tessellation.

Delaunay Triangulation: Delaunay Triangulation (Musin, n.d.) is a technique that connects a set of points in a plane, forming a triangulation such that no point is inside the circumcircle of any triangle. As shown in Figure 9, this results in a set of edges with no overlaps that maximize the minimum inside angles of the triangles. This method has been employed in various studies, such as the one by (Lu et al., 2014), where they used Delaunay triangulation to define the connectivity of subgraphs of cells in a WSI. The Voronoi Tessellation is dual to the Delaunay Triangulation[5].

[5] The Voronoi Tessellation and Delaunay Triangulation are duals of each other. This means that if we take the Voronoi Tessellation of a set of points in the plane and define edges

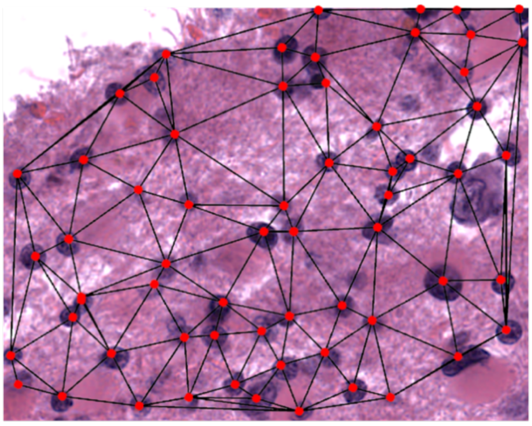


Figure 9. Graph generated with Delaunay triangulation connecting the nuclei of the cells, superposed on the original WSI.

***K*-Nearest Neighbours (*K*NN):** *K*NN is an algorithm that computes the distance between all pairs of nodes in a graph and creates new relationships between each node and its *k* nearest neighbours (Bhatia & Vandana, 2010). In the context of WSIs, *K*NN can be used to connect nodes (i.e., cellular nuclei) based on their distance in the WSI. This approach can help create a graph representation of the WSI, where nodes are connected based on their proximity.

Given the geometric duality between Voronoi Tessellation and Delaunay Triangulation (Aurenhammer et al., 2013), applying either of these techniques to the same set of points (i.e., the positions of cell nuclei) will yield identical graphs in terms of connectivity. This is because the edges that form between nodes, that model cell interaction in the graphs, are determined by the same spatial relationships, regardless of the technique. Consequently, while both methods have their distinct advantages and applications, in the context of defining edges for our cell-graphs, they are effectively interchangeable, and therefore we will only be using Delaunay Triangulation, as it is the most direct approach to compute the edges.

In Figure 10, we can observe the differences in the graphs created with Delaunay triangulation and *K*NN. Note that the graph created with *K*NN is a directed graph, the edges have a source and a target that cannot be interchanged. This is due to the fact that the relationship in *K*NN is not symmetrical, which means a node's *K*-nearest neighbours do not necessarily count that node among their own *K*-nearest neighbours.

between the nodes if their regions are adjacent, we obtain a graph that is identical to the Delaunay Triangulation of that set of points. The duality relationship between Voronoi diagrams and Delaunay triangulations is a fundamental concept in computational geometry (Aurenhammer et al., 2013).

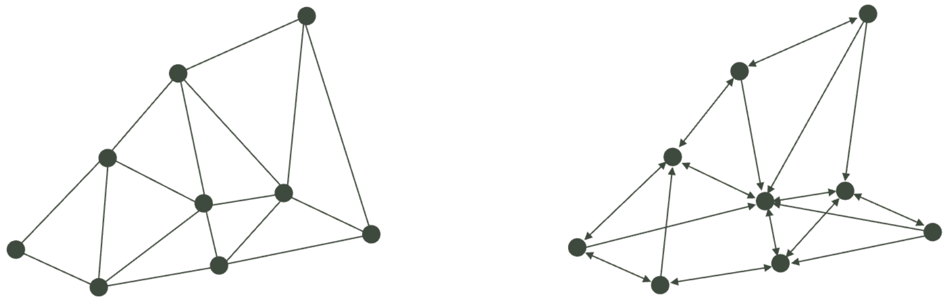


Figure 10. Example of the differences in Edge Definition with the application of Delaunay Triangulation (left) and KNN with K = 3 (right).

3 Data

3.1 Nuclei Representation

In the context of graph-based learning, the nuclei features play a crucial role in encoding cellular tissue data as graphs. The data provided by NCMM for this project (see Table 1) consisted of 5 different sets of nodes from 2 different tumour categories, that had been derived from WSIs, and included the following features:

Nuclei Size: Area enclosed by the nuclear membrane. Nuclei size can be a valuable node feature in cell-graphs, as it provides insights into the morphological characteristics of cells (Wang et al., 2022).

Nuclei Position: X and Y coordinates of the nuclei's centroid in the WSI. In Figure 11, we can see the nuclei of a tumour's WSI represented as dots in a plane given their X and Y coordinates.

Hu Moments: Hu moments are a set of image descriptors that capture the shape and texture characteristics of an object (*Hu Moments*, 2019). They are a set of seven numbers calculated using central moments that are invariant to image transformations[6]. Hu Moments can be used to encode the nuclei shape. The nuclei shape can be described by a feature vector of seven values that capture and quantify the shape of the object in an image, giving a quantification on how similar two shapes are (Satya Mallick & Krutika Bapat, 2018). As the main distinction between High- and Low-Grade Gliomas is the differentiation of their cells (Section 1.1), Hu moments can be useful in tumour classification.

An example of how the original datasets including the node features mentioned above can be found in Table 2.

[6] Note: While most Hu moments are invariant to image transformations, the 7th Hu moment is an exception as it changes its sign if the image is reflected.

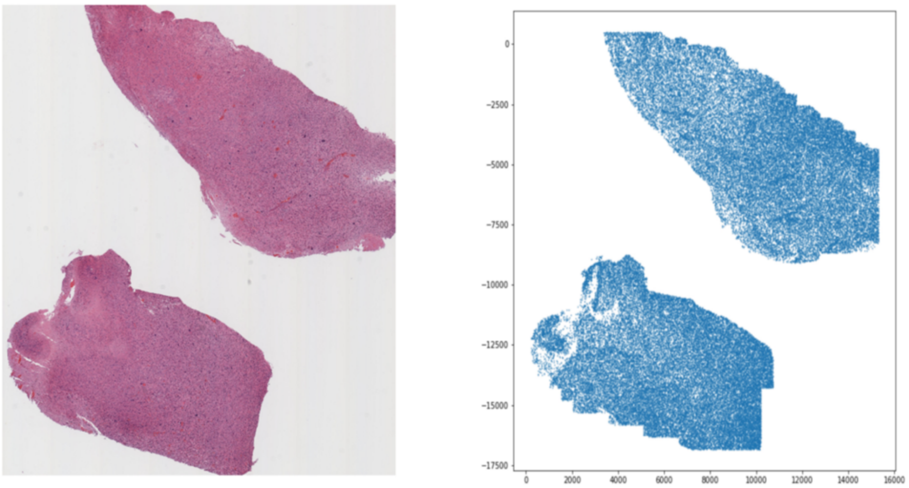


Figure 11. Left: Original High-Grade Glioma WSI. Right: Position of the nuclei derived from the High-Grade Glioma WSI. figures provided by NCMM.

Name in NIH Data Portal	Nodes	Label	NIH Link
TCGA-06-0187-01Z-00-DX3	71705	GBM	NIH link
TCGA-06-0238-01Z-00-DX1	16322	GBM	NIH link
TCGA-DU-7018-01Z-00-DX1	60826	LLG	NIH link
TCGA-HT-7479-01Z-00-DX1	4687	LLG	NIH link
TCGA-HT-8563-01Z-00-DX1	373211	LLG	NIH link

Table 1. Summary of the datasets provided by NCMM. This data was generated from WSIs and contains the node features shown in Table 2. Links to the National Health Institute slide viewer in the left column for quick visualization of the WSIs.

NODE_ID	COORD_X	COORD_Y	SIZE	HU_1	HU_2	HU_3	HU_4	HU_5	HU_6	HU_7
0	220	-12629	214.87	0.1606	0.0002	2.103E-06	6.954E-09	-5.593E-16	-1.143E-10	-6.293E-16
1	460	-12253	172.26	0.1712	0.0033	0.00013	2.757E-06	-5.221E-11	-1.546E-07	-7.775E-12
2	378	-12379	122.25	0.1628	0.0006	2.850E-05	1789E-07	4.028E-13	3.829E-09	3.234E-14

...

Table 2. Example of nuclei features provided by NCMM. This represents the starting point in terms of data for this study.

3.2 Data to Graphs

During the node feature definition, we decided to omit the X and Y coordinates from the feature set of the nodes. This decision may appear counter-intuitive as the position of a nucleus can provide valuable information about its environment. However, it's crucial to remember that these coordinates will already serve a role in the edge definition process, where we used them to establish spatial relationships between nuclei (see Section 2.3).

Including the coordinates as features could lead to redundancy in the graph model, which could, in turn, potentially confuse the learning process of our GNNs. More importantly, as the GNN models are designed to be invariant to the permutation of the nodes, including the absolute spatial coordinates as features would violate this principle and could adversely affect the learning process.

Therefore, in our graphs, each node represents a nucleus characterized by its size and the seven Hu moment invariants. These features provide rich information about the individual nuclei while ensuring that the graph model appropriately captures the spatial relationships between them through the edges. This approach allows us to maintain the translational invariance of our model while preserving the critical spatial information within the edge structure.

Upon establishing the nodes of our graphs, the next crucial step involves defining the edges. This process relies on the techniques delineated in Section 2.3, specifically Delaunay Triangulation and K-Nearest Neighbours.

4 Methodology

The primary objective of this study is to assess the effectiveness of GNNs in classifying brain tumours using graphs generated from WSIs. This task is particularly challenging due to the complex and heterogeneous nature of brain tumours, which necessitates the extraction and analysis of intricate relationships within the tumour microenvironment.

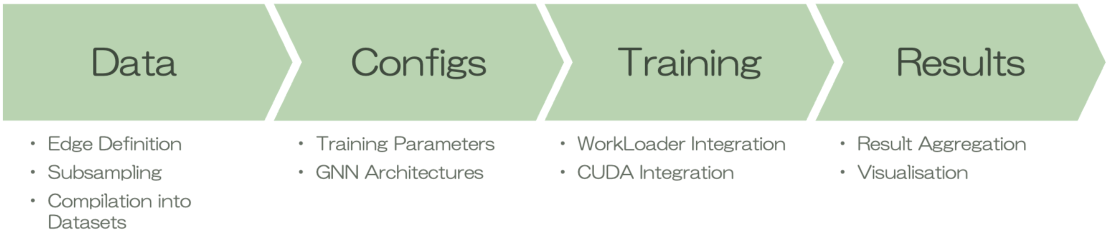
However, tracking progress and evaluating task-specific performances is often challenging due to the wide variety of architectures available as well as graph entity representation optimizations for such specific fields like the one we are involved in.

For that reason, we have built a framework that intends to help with that (see Figure 12). This framework brings forward all the necessary steps to complete the tumour classification task that was proposed by NCMM. Furthermore, when developing the infrastructure, we have prioritized ease-of-use, futureproofing, and the facilitation of both breadth and depth in research.

While our infrastructure builds upon the foundation provided by GraphGym, we have adapted and extended this platform to suit the specific requirements of brain tumour classification.

To test our framework, we decided to explore different state-of-the-art graph generation pathways from histopathological images and assess how they influence the models' performances. Furthermore, we trained GNN models on these graphs, learning to aggregate and transform the feature information from the nodes and their local neighbourhoods into global graph representations for downstream classification.

We allow the performance of the GNNs to be evaluated using several metrics, namely AUC, F1 score, accuracy, precision, and recall. These metrics provide a comprehensive assessment of the model's performance, capturing its ability to distinguish between different classes of brain tumours.


Data
- Edge Definition
- Subsampling
- Compilation into Datasets

Configs
- Training Parameters
- GNN Architectures

Training
- WorkLoader Integration
- CUDA Integration

Results
- Result Aggregation
- Visualisation

Figure 2. Overview of the framework's pipeline.

Through this framework, we aim to provide an evaluation of GNNs for brain tumour classification, shedding light on their strengths and weaknesses, and paving the way for future research in this important area. Furthermore, by comparing the performance of GNNs across different edge construction methodologies, we seek to understand the influence of the graph structure on the model's performance, providing insights that could guide the design of more effective GNNs for brain tumour classification.

4.1 Parameter budget

In the design of our benchmarking framework, we have carefully considered the parameter budget. Our goal was not to find the optimal hyper-parameterization for brain tumour classification using GNNs, as this would be computationally expensive and beyond the scope of our project. Instead, our aim was to compare the performance of different models and their building blocks within a low-budget parameter limitation.

This approach allows us to conduct a broad comparison of different models and configurations, providing valuable insights into their relative strengths and weaknesses. It also enables us to explore the impact of different edge-definition mechanisms on the performance of the GNN models.

While this low-budget parameter limitation restricts the complexity of the models and configurations we can explore, it also ensures that our benchmarking process is feasible given the computational and storage demands of processing the resulting graph structures. This approach aligns with our design considerations for ease-of-use and futureproofing, as it allows for a wide range of researchers to replicate our experiments and build upon our work.

In this context, our parameter budget serves as a practical constraint that guides our benchmarking process, ensuring that it is both rigorous and feasible. In Table 3, we can see a summary of the different parameters that we decided to benchmark, including their corresponding values.

4.2 Framework's Pipeline

4.2.1 Data Pre-processing

It is unclear what theoretical methodologies can define best the edges of our graphs in the context of brain oncology. Furthermore, the processing of the graphs resulting from WSIs requires extensive CPU and/or GPU resources due to their size and complexity, which can be a limiting factor in many research settings.

To address these challenges, we have taken several design considerations into account in the creation of our datasets. Our primary strategy has been to reduce the complexity of the problem by performing subsampling on the original WSI graphs. This approach allows us to maintain the essential structure and information of the graphs while significantly reducing the computational demands of processing them.

The induced subgraphs are then compiled into datasets (see Table 5). These present the different edge-definition mechanisms that we have considered for this task (Section 2.3). In Table 3, we can see an example of the data formats compiled into the datasets.

Subgraph ID	Node Features				Edge List			Edge Features		Label
	Node ID	Size	HU_1	...	Edge ID	Source	Target	Edge ID	Weight	
0	0	341.40	0.16	...	0	0	2	0	56.92	GBM
	2	556.59	0.16		1	0	10	1	36.71	
	10	222.14	0.18		2	0	18	2	20.0	
	18	243.76	0.16		3	0	23	3	56.0	
	23	196.5	0.17		7	2	18	7	70.25	
	...				44	10	18	44	34.0	
					63	10	23	63	49.68	
					104	18	23	104	69.85	
...										

Table 3. Example of a subgraph inside a PyG dataset with Delaunay Triangulation. For each of the subgraphs that have been induced, we have its node features, edge list and edge features, as well as their label.

Training our models on these datasets allows for a feasible training time given the resources provided for this project. It also enables us to systematically evaluate the impact of different edge-definition mechanisms on the performance of the GNNs, providing valuable insights for the design of future models and experiments.

4.2.2 Experiment definition

Our framework allows for easy GNN and training configuration within a single file by just specifying the parameters and their desired values. Moreover, it facilitates the generation of a grid of experiments by setting lists of values for each of the parameters. This capability simplifies the setup of proof-of-concept experiments, providing initial insights into potential model configurations for specific tasks. Additionally, it streamlines the fine-tuning process, enabling users to experiment with various hyper-parameters. It allows researchers to refine and optimize existing configurations, making this framework a versatile tool for both exploratory and optimization tasks.

4.2.3 Training

In this study, we've developed a comprehensive framework that is designed to streamline the process of graph generation, training, and testing of GNNs. A crucial component of this framework is its integration with both the Slurm Task Manager and CUDA, which offers an efficient and scalable approach to managing and executing computational tasks.

The Slurm Workload Manager (*Slurm Workload Manager - Overview*, 2021) plays a pivotal role in handling the computational tasks in our framework. Slurm is a highly configurable open-source workload manager used in a large number of high-performance computing clusters around the world. It provides a robust framework for job scheduling, queuing, prioritization, and resource management. In the context of our work, Slurm enables the efficient management of multiple computational tasks, such as generating graphs and training GNN models, allowing for easy scalability and improved productivity. It ensures that computing resources are optimally utilized and that tasks are executed in an orderly and efficient manner.

Complementing the capabilities of Slurm, our framework also leverages the power of CUDA (Compute Unified Device Architecture) (Nickolls et al., 2008), a parallel computing platform and application programming interface model created by Nvidia. CUDA allows for direct access to the virtual instruction set and memory of the parallel computational elements in GPUs. By integrating CUDA, our framework can execute GNN training on Nvidia GPUs, significantly accelerating the training process and enabling the handling of large and complex datasets.

The synergy between the Slurm Task Manager and CUDA in our framework allows for efficient scheduling and rapid execution of computational tasks. This integration facilitates handling high volumes of data and complex GNN architectures, significantly accelerating the pace of research and providing a robust and scalable infrastructure for exploring the effectiveness of GNNs in brain tumour classification.

4.2.4 Results Aggregation and Visualization

Finally, this framework stores epoch-resolution results with 5 different performance metrics and their standard deviations so researchers can use them for their specific requirements. Moreover, the system automatically aggregates the results after the training process concludes into easy-to-handle csv files.

Furthermore, it also provides visualization tools that facilitate visual inspection of the results and allow for an easy comparison of how the different parameter configurations affect the performance of the models.

4.3 Datasets

This section presents an overview of the datasets generated during our study. A central part of our research involves defining the edges of cell-graphs, a process that plays a crucial role in modelling the intricate interactions between tumour cells in brain cancer histopathology images. In our quest to identify the optimal edge-definition technique, we have generated multiple datasets using four different approaches: Delaunay Triangulation, and K-Nearest Neighbours (KNN) with K set to 5, 10, and 15 (Table 4).

Each method offers a unique way to capture the complex spatial relationships between the cells, and the choice of method can significantly impact the performance of the GNN models. By comparing the results obtained from these different datasets, we aim to identify the most effective edge-definition technique and to contribute to the development of standard methods for comparing and evaluating such techniques.

Method	Subgraphs	Avg. Nodes	Avg. In-degree	Avg. Edges	Avg. Edge Distance
Delaunay	100	500	2.80	1401.23	61.69
KNN (K=5)	100	500	4.57	1142.30	36.71
KNN (K=10)	100	500	9.64	2409.45	52.41
KNN (K=15)	100	500	14.51	3626.33	64.41

Table 4. Summary of the datasets used to train the GNN models.

4.4 GNN Models

In Table 5 we can observe the different GNN configurations that we decided to benchmark.

Parameters	Values
GNN Layer Type	GAT, GCN, GIN, GraphSAGE
Residual Connections	SkipConcat, SkipSum, Stack
Dropout Rate	0.0, 0.1
Input Layers	3, 5
Message Passing Layers	5, 7, 9
Task-Based Layers	3, 5

Table 5. Summary of the different parameters benchmarked and their values.

5 Results

In this section we provide an overview on the results of the benchmark of the GNN configurations previously introduced (Section 4.3). Note that in order to evaluate and compare both the edge-definition methodologies and the performance of the GNN configurations that we propose, we trained the models on the different datasets that we created (see Table 4).

5.1 Edge-Definition Methodologies

In Figure 13, we can see the AUC distribution, average rank, and rank distribution of the different edge-definition approaches. Note that all KNN variants ended up with better results than Delaunay, with the best performance achieved by $K = 10$.

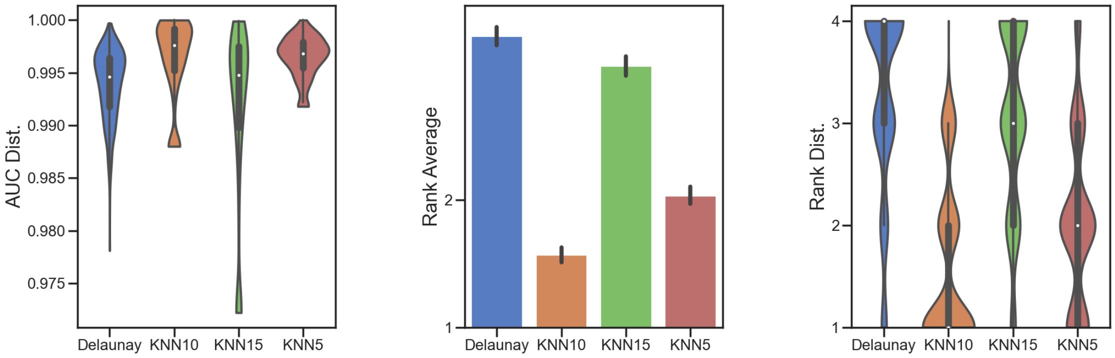


Figure 3. Benchmarking results comparing the performances achieved by the different edge-definition mechanisms proposed. Left: Distribution of the AUC values (Higher is better). Middle: Rank Average (Lower is better). Right: Distribution of the Ranks obtained (Lower is better).

5.2 GNN Architectures

In the following sections, we discuss the results obtained from the GNN models introduced in Section 4.4, focusing on the performance as well as their complexities in terms of number of parameters.

While we computed several performance metrics, including F1 score, accuracy, precision, and recall, our evaluation is primarily based on the Area Under the Receiver Operating Characteristic Curve (AUC), which is usually preferred for binary classification. This is the standard approach suggested by NCMM for evaluating model performance, particularly given the balanced nature of our datasets. Precision, recall, and F1-score, while valuable, do not offer a comprehensive overview of model performance in our context.

5.2.1 Model Performances

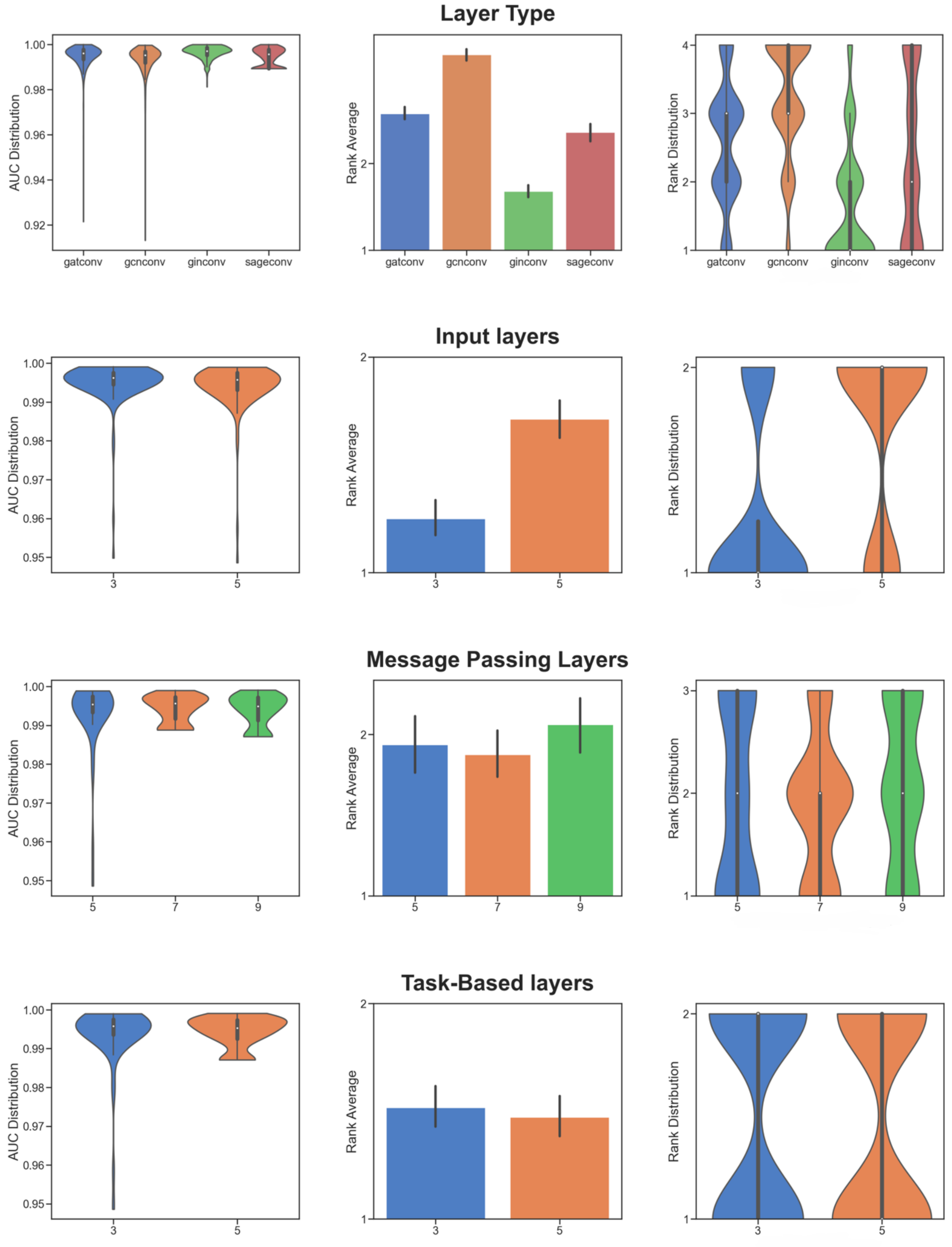


Figure 14, part 1. Performance of the models measured by the AUC metric. Left: Distributions of the AUC obtained by the different parameters (Higher is better). Middle: Average ranking of the models' performances for the different parameter values against their counterparts (Lower is better). Right: Distribution of the rankings obtained by the models' performances for the different parameter values against their counterparts (Lower is better).

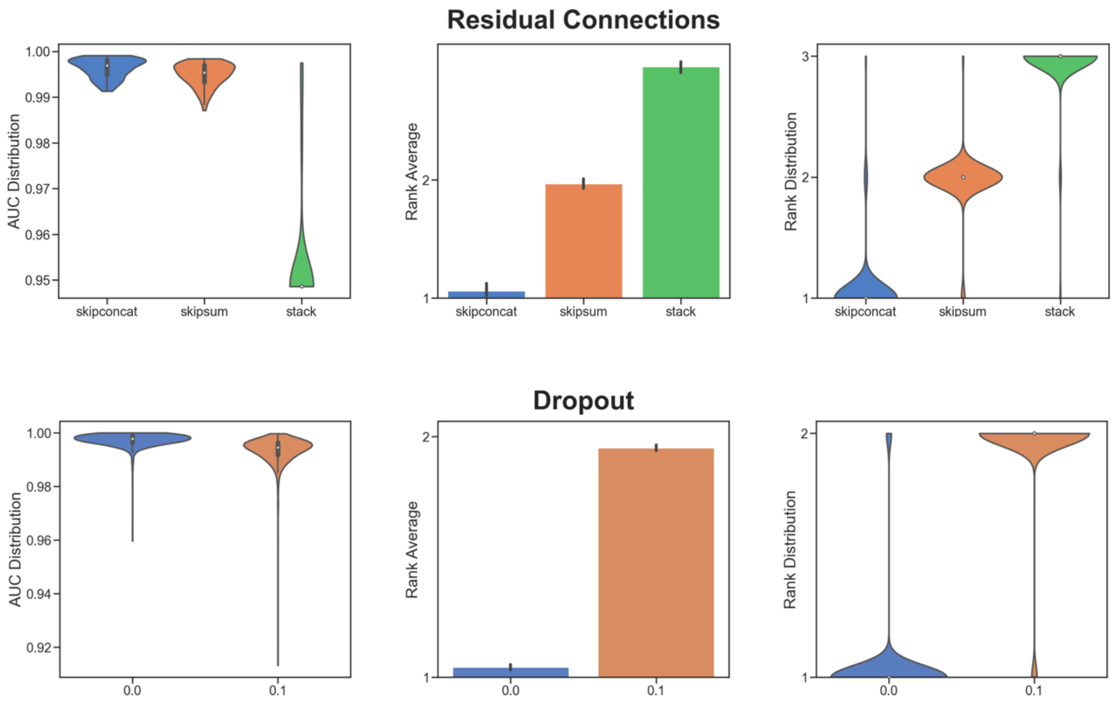


Figure 14, part 2.

An examination of the performance results outlined in Figure 14 reveals that the GIN layer type consistently obtained the highest AUC scores across all datasets, thus highlighting its efficacy in the brain tumour classification task.

Our findings suggest that configurations utilising a lesser number of input layers (specifically, three layers) generally superseded their five-layer counterparts in AUC performance. This seemingly contradicts the expectation that higher complexity should yield better results, instead suggesting that simpler models can adeptly encapsulate the necessary features for precise classification.

A close look at the results obtained from experiments with various numbers of message passing layers (five, seven, and nine layers) discloses that they returned AUC scores of near parity.

Regarding the task-based layers, a noteworthy observation was that augmenting the number of layers from three to five did not engender substantial improvements in AUC.

A clear enhancement in model performance was discernible with the addition of residual connections, as opposed to their absence (‘stack’). Among these, the 'skipconcat' residual connection stood out by delivering the best overall results.

The analysis also revealed that a dropout rate of 0.0 yielded the highest AUC among the various configurations tested.

5.2.2 Model Complexities

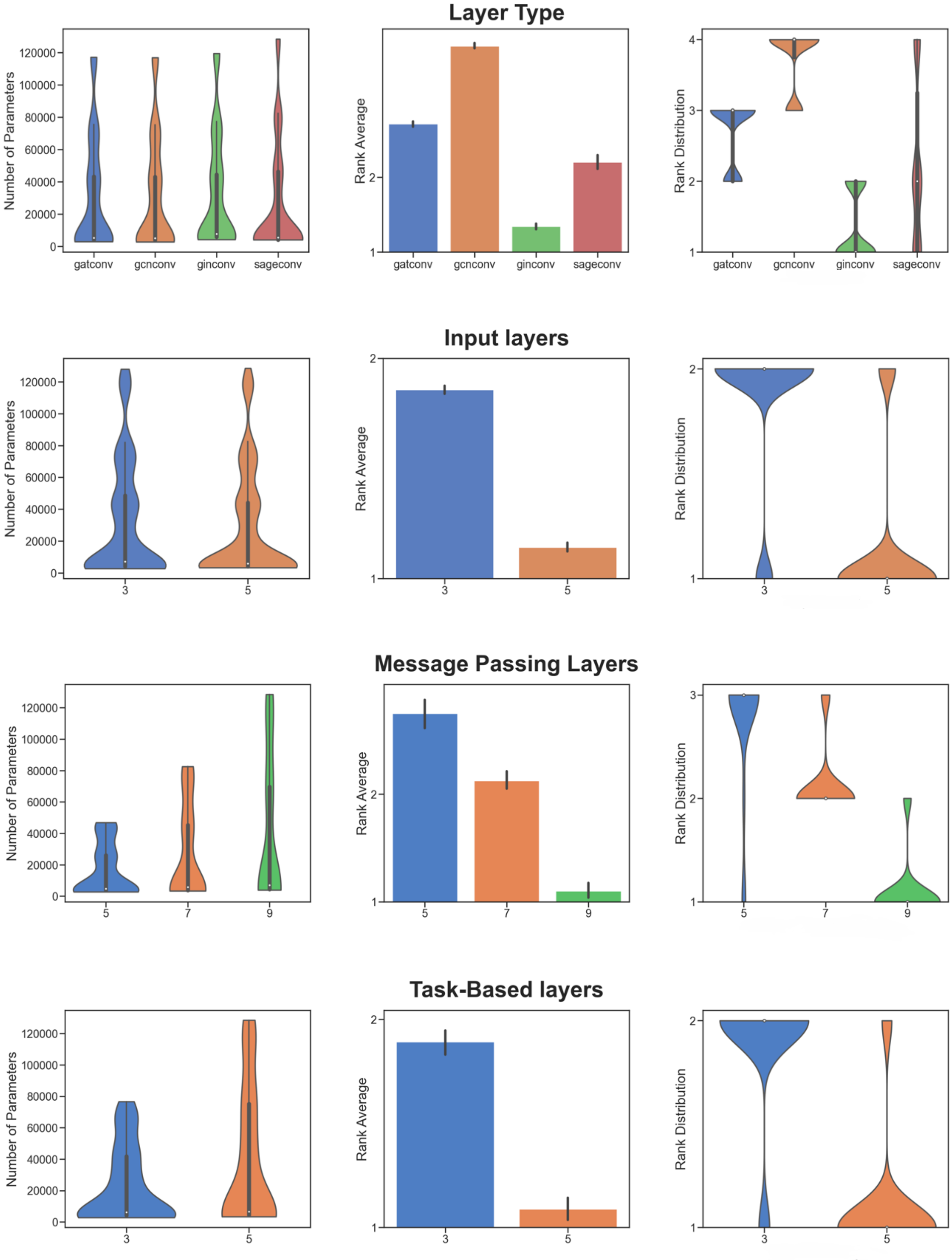


Figure 15, part 1. Complexity of the models measured by their number of parameters. Left: Distributions of the model complexities obtained by the different parameters (Lower is better). Middle: Average ranking of the models' complexities for the different parameter values against their counterparts (Higher is better). Right: Distributions of the rankings obtained by the models' complexities for the different parameter values against their counterparts (Lower is better).

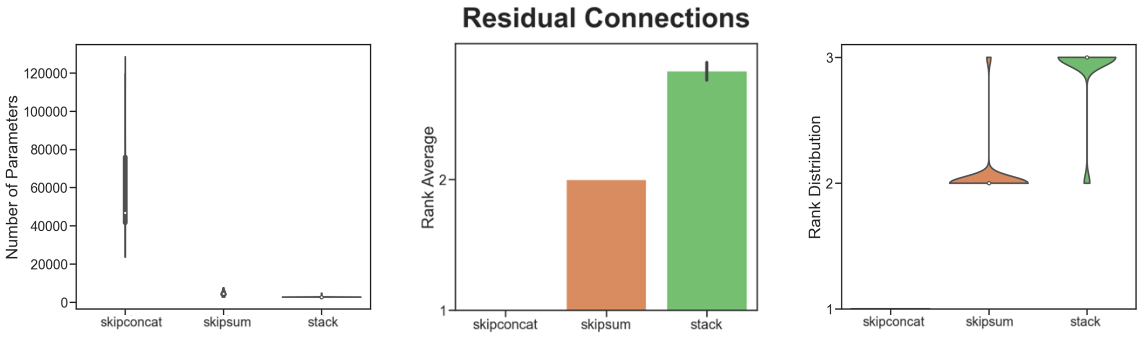


Figure 15, part 2.

Indeed, the analysis of model complexity revealed the GIN layer type to be among the most complex models in terms of parameter count. This additional complexity is inherent to the design of the GIN layer type and contributes to its unique ability to capture intricate patterns within the data.

Despite its complexity, the GIN layer type consistently exhibited superior performance, earning the highest AUC scores. This correlation between complexity and high performance in this particular case suggests that the GIN layer type effectively harnesses its larger number of parameters to derive meaningful insights, a feat not readily achieved by simpler models.

The analysis of model complexity, gauged by the parameter count, unveiled some noteworthy patterns. Configurations with fewer input layers (specifically, three) despite being less complex, did not compromise on performance, challenging the conventional wisdom that more parameters inherently mean superior model quality.

Variations in the number of message passing layers (five, seven, and nine layers) essentially represent alterations in model complexity. However, the impact on AUC scores seemed marginal, suggesting that increasing model complexity, as indicated by the number of parameters, may not always result in performance improvements.

In the context of task-based layers, the increase in complexity from three to five layers did not offer any significant advantage, suggesting that a simpler configuration with fewer parameters could suffice for the classification task.

The introduction of residual connections added a layer of complexity to the models, particularly the 'skipconcat' connection, which nonetheless produced the best overall results. This instance highlights one of the few scenarios where increased complexity correlated with improved performance.

6 Discussion

6.1 Edge Definition Approaches

KNN with $K = 10$ achieved the best performance. Contrary to our initial intuition that Delaunay Triangulation, being a dual to Voronoi Tessellation, would excel in capturing cell interaction, our results showed that all variants of the KNN method outperformed Delaunay Triangulation. Interestingly, the best performance was achieved with K=10, a variant that deviates substantially from the boundary adjacency of cells, which we expected Delaunay to capture closely.

This outcome prompts an interesting consideration, particularly considering the points raised by Nair et al., 2022. They noted potential issues with the KNN method, namely that KNN-based edge generation is not invariant with respect to node ordering in a graph, potentially generating non-isomorphic cell-graphs from the same tissue image. This lack of invariance might theoretically impact the model's robustness to tissue orientation and complicate the learnability of histological features.

However, different tumours might have distinct distribution characteristics, with some exhibiting areas where cells form ring-like or palisading structures, or areas of high mitotic activity. These unique patterns could potentially be captured more effectively by the more flexible KNN, as it doesn't enforce reciprocity and might better encapsulate clusters of cells.

In our study, despite the theoretical concerns, KNN demonstrated robust performance in the task of brain tumour classification, outperforming the more theoretically appealing Delaunay Triangulation. This result underscores the importance of empirical evaluation in complement to theoretical analysis, and suggests that there is value in further exploring the balance between theoretical soundness and empirical performance in the application of GNNs for histopathological image analysis.

6.2 GNN Models' Performances

GIN Layer Type outperforms other proposed types. The GIN layer type ended up generating better performances than its counterparts. This could be due to the fact that this layer type was specifically designed to tackle the graph isomorphism problem (see Section 2.2.2.4). However, the complexity of this problem increases with the amount of graph categories to discern from. In this project we have been trying to classify our graph into two categories – high- and low-grade gliomas – achieving better results than its counterparts. Nevertheless, when scaling this classification problem to more tumour categories, we could potentially see other layer types achieving a better trade off in terms of model complexity vs. performance.

For Input Layers, less is more. The 3-layers variants performed better than the 5-layers variants. This could be explainable with the amount of information that could be lost with the initial embedding generation.

Similar results regardless the amount of Message Passing Layers. The number of Message Passing Layers did not affect the classification performance significantly, as the 5-, 7- and 9-layer variants showed similar results. While it might seem intuitive to add more layers to potentially increase learning capacity, it's important to consider the implications of model complexity. More layers mean a more complex model, which in turn can lead to an increased risk of overfitting, especially when dealing with limited data.

Increasing the number of task-based layers did not improve performance. There were no significant differences when increasing the depth of the task-based layers that perform the classification of the graph embedding generated by the message passing layers. This could be explained by the fact that binary classification is a relatively less complex task compared to multi-class classification or regression. Therefore, having more task-based layers may not significantly improve the performance because the task might not require a high level of complexity or abstraction that additional layers would provide.

Residual connections improve model performances. Adding residual connections between layers allowed for a more efficient training, allowing for better performances in the fixed number of epochs of our experiments.

No Dropout achieved best performance. Reducing the Dropout to zero tends to result in overfitting of the models and should usually be avoided as it can lead to poor validation performances and generalization of the models outside the training datasets. However, we accounted for that by adding Batch Normalization operations (see Section 2.2.2.5) that help mitigate the effects of removing the Dropout. This allows for a faster and more efficient training while also achieving better performances for the fixed number of epochs that we set for our experiments.

6.3 Framework Design Choices

The design choices for our GNN benchmarking framework reflect our goal of enabling reliable prototyping for GNN research applied to WSIs. Acknowledging the brief duration of this project and the considerable computational time required for training (which can extend to numerous hours for each experimental run), we deemed it necessary to establish certain boundaries. These boundaries affect the complexity of the models, the size of the datasets, and the quantity of models that could feasibly be tested within the constraints of this project.

Firstly, our decision to use subsampled datasets allows us to manage the computational and storage demands of processing the huge graph structures that were generated from the data originally provided. This approach enables us to conduct a broad comparison of different models and configurations, providing valuable insights into their relative strengths and weaknesses.

Secondly, our coding infrastructure, with its integration of Slurm task manager and CUDA environments, ensures efficient scheduling and management of computational tasks, leveraging GPU acceleration for improved performance. This design choice facilitates the handling of the computationally intensive task of training GNNs on graph-structured data.

Thirdly, our decision to limit the parameter budget reflects our goal of comparing the models and their building blocks within a low-budget parameter limitation, rather than finding the optimal hyper-parameterization for brain tumour classification using GNNs. This approach allows us to conduct a broad comparison of different models and configurations, providing valuable insights into their relative strengths and weaknesses.

However, it is true that the variability in the number of parameters of the models and their run/training times makes it a bit difficult to assess their scalability. This is an inherent challenge in benchmarking GNNs, and one that future research may help to address.

In our study, we adopted a subsampling approach to manage the original graphs, a decision driven by two primary factors. Firstly, subsampling effectively curbs computational demands, facilitating more efficient processing. Secondly, due to the limited volume of available data, subsampling served as a practical strategy for data augmentation, enabling us to generate a more diverse set of samples for model training.

However, it's crucial to note that while subsampling offers these benefits, it also introduces certain challenges. Specifically, the process can result in the loss of some information, which could potentially impact the performance of the GNNs. Additionally, if the sizes of the original graphs and subgraphs are not carefully managed, the risk of overfitting increases, which could affect the generalisation capabilities of the GNNs. Consequently, a careful balance was necessary to maximize the advantages of subsampling while minimizing its potential drawbacks.

7 Conclusions

In conclusion, this study presents a significant stride towards improving brain tumour classification, a complex task with substantial implications for patient diagnosis and treatment in neuro-oncology. Our research aimed to identify the most effective combination of edge-definition techniques and GNN architectures.

The results reveal that the KNN approach with K=10 is the optimal edge-definition technique, and the Graph Isomorphism Network (GIN) layer type outperforms others in this context. These findings offer valuable guidance to researchers seeking to further explore the potential of histological image-based tumour diagnosis using GNNs.

We have also introduced a comprehensive framework for evaluating the effectiveness of GNNs for brain tumour classification. This framework illuminates the strengths and weaknesses of various GNN models and edge construction methodologies, offering insights that could inform the design of more effective GNNs for this critical task. Moreover, by comparing the performance of GNNs across different edge construction methodologies, our study elucidates the impact of graph structure on model performance.

The implications of this research extend beyond brain tumour classification, as the framework and insights gained should be generalizable to other areas in oncology. By providing a benchmark for model evaluation and comparison, this study contributes to the standardization of methodologies in the field and paves the way for future research in GNN-based cancer diagnosis.

8 Future Work

Looking forward, we anticipate that the insights gained, and the framework developed in this study will drive further advancements in the application of GNNs in oncology. Future research could also explore the integration of additional biological and clinical variables into the graph models, potentially enhancing their predictive power and clinical utility. Through these continued efforts, we aspire to further improve the accuracy and efficiency of cancer diagnosis, ultimately contributing to improved patient outcomes in neuro-oncology and beyond.

While our research has taken significant strides in advancing the application of Graph Neural Networks (GNNs) for brain tumour classification, there remain several avenues for future exploration. One particularly important direction is the need for greater explainability in our models. As we continue to refine the accuracy of our GNNs, it's equally crucial that we gain a deeper understanding of the factors that primarily drive their decisions.

Such insights could not only improve our confidence in the model's predictions but also potentially reveal novel insights about the characteristics and interactions of nuclei that are most indicative of specific tumour types. This could pave the way for more targeted and effective treatment strategies in neuro-oncology.

To this end, we plan to leverage the suggestions put forth by Sebastian Waszak and Birgit Kriener. Their proposition involves training the models on graphs where the edge weights are set to unitary values instead of distances, or even removing node features entirely. Such approaches could provide a clearer picture of whether it's the nuclei characteristics or their interactions that are the primary drivers of the models' decisions in classifying the tumour types.

By isolating these factors, we aim to dissect the contributions of individual nodes and edges to the final classification decision. This could lead to a more nuanced understanding of how the spatial distribution and relationships of nuclei influence the diagnosis of brain tumours. Such insights could potentially inspire new questions and hypotheses in neuro-oncology, opening exciting new directions for future research.

Another promising direction for future research involves experimenting with different distance metrics for the K-Nearest Neighbours (KNN) edge-definition approach, which our study identified as the best performing technique.

Currently, our KNN approach relies solely on the XY coordinates of the nodes to determine their 'closeness'. While this spatial proximity is important, incorporating additional information into the distance metric could potentially enrich the edges of our cell-graphs. For instance, we could implement a distance metric that considers the node features, such as the size and the seven Hu moment invariants of the nuclei. With such a metric, nodes with similar features would be 'closer' to each other, even if they are not the nearest in terms of spatial coordinates.

This approach could enable our model to capture not only the spatial relationships between nuclei, but also their similarities in terms of physical and morphological characteristics. By creating edges that reflect these multi-dimensional relationships, we could potentially enhance the performance of our GNNs and gain more nuanced insights into the factors that drive their decisions.

Another promising avenue of research could be to explore the potential of making the edge-definition mechanism a learnable parameter within the GNN itself. Currently, the edges are defined prior to training based on a pre-determined method, such as KNN or Delaunay Triangulation. However, if the edge-definition process could be learned during training, the network might be able to better adapt to the specific characteristics of the data. This could potentially result in a more flexible model capable of capturing more complex and nuanced relationships between nodes. Indeed, the ability to learn and adapt the graph structure during training could open up new possibilities for enhancing the performance of GNNs in brain tumour classification, as well as other applications.

9 References

Ahmedt-Aristizabal, D., Armin, M. A., Denman, S., Fookes, C., & Petersson, L. (2022). A survey on graph-based deep learning for computational histopathology. *Computerized Medical Imaging and Graphics*, *95*. https://doi.org/10.1016/J.COMPMEDIMAG.2021.102027

Al-Hussaini, M., & Al-Hussaini, M. (2013). Histology of Primary Brain Tumors. *Clinical Management and Evolving Novel Therapeutic Strategies for Patients with Brain Tumors*. https://doi.org/10.5772/52356

Aurenhammer, F. (1991). Voronoi diagramsa survey of a fundamental geometric data structure. *ACM Computing Surveys (CSUR)*, *23*(3), 345–405. https://doi.org/10.1145/116873.116880

Aurenhammer, F., Klein, R., & Lee, D.-Tsai. (2013). *Voronoi Diagrams and Delaunay Triangulations*. World Scientific.

Bhatia, N., & Vandana. (2010). Survey of Nearest Neighbor Techniques. *IJCSIS) International Journal of Computer Science and Information Security*, *8*(2). https://arxiv.org/abs/1007.0085v1

Bock, M., Tyagi, A. K., Kreft, J.-U., & Alt, W. (2009). *Generalized Voronoi Tessellation as a Model of Two-dimensional Cell Tissue Dynamics*. http://arxiv.org/abs/0901.4469

Brain tumours - NHS. (2023, June 12). https://www.nhs.uk/conditions/brain-tumours/

Chan, T. H., Cendra, F. J., Ma, L., Yin, G., & Yu, L. (n.d.). *Histopathology Whole Slide Image Analysis with Heterogeneous Graph Representation Learning*. Retrieved July 8, 2023, from https://github.com/HKU-MedAI/WSI-HGNN.

Charles, N. A., Holland, E. C., Gilbertson, R., Glass, R., & Kettenmann, H. (2011). The brain tumor microenvironment. *Glia*, *59*(8), 1169–1180. https://doi.org/10.1002/GLIA.21136

Cheng, L., Wu, Q., Guryanova, O. A., Huang, Z., Huang, Q., Rich, J. N., & Bao, S. (2011). Elevated invasive potential of glioblastoma stem cells. *Biochemical and Biophysical Research Communications*, *406*(4), 643–648. https://doi.org/10.1016/J.BBRC.2011.02.123

David Reardon, M. (2019, June 4). *What Makes a Brain Tumor High-Grade or Low-Grade?* . https://blog.dana-farber.org/insight/2018/04/makes-brain-tumor-high-grade-low-grade/

Doi, K. (2007). Computer-aided diagnosis in medical imaging: Historical review, current status and future potential. *Computerized Medical Imaging and Graphics*, *31*(4–5), 198–211. https://doi.org/10.1016/J.COMPMEDIMAG.2007.02.002

Doolittle, N. D. (2004). State of the science in brain tumor classification. *Seminars in Oncology Nursing*, *20*(4), 224–230. https://doi.org/10.1016/J.SONCN.2004.07.002

Douglas, B. L. (2011). *The Weisfeiler-Lehman Method and Graph Isomorphism Testing*. https://arxiv.org/abs/1101.5211v1

Hamilton, W. L., Ying, R., & Leskovec, J. (2017). Inductive Representation Learning on Large Graphs. *Advances in Neural Information Processing Systems*, *2017-December*, 1025–1035. https://arxiv.org/abs/1706.02216v4

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep Residual Learning for Image Recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2016-December*, 770–778. https://doi.org/10.1109/CVPR.2016.90

Hu Moments. (2019, July 21). https://cvexplained.wordpress.com/2020/07/21/10-4-hu-moments/

Ioffe, S., & Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *32nd International Conference on Machine Learning, ICML 2015*, *1*, 448–456. https://arxiv.org/abs/1502.03167v3

Johann Huber. (2020, November 6). *Batch normalization in 3 levels of understanding | by Johann Huber | Towards Data Science*. Batch Normalization in 3 Levels of Understanding. https://medium.com/towards-data-science/batch-normalization-in-3-levels-of-understanding-14c2da90a338

Kipf, T. N., & Welling, M. (2016). Semi-Supervised Classification with Graph Convolutional Networks. *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*. https://arxiv.org/abs/1609.02907v4

Levy, J., Haudenschild, C., Barwick, C., Christensen, B., & Vaickus, L. (2021). Topological Feature Extraction and Visualization of Whole Slide Images using Graph Neural Networks. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, *26*, 285. https://doi.org/10.1142/9789811232701_0027

Lu, Z., Luo, Z., Zheng, H., Chen, J., & Li, W. (2014). A Delaunay-Based Temporal Coding Model for Micro-expression Recognition. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *9009*, 698–711. https://doi.org/10.1007/978-3-319-16631-5_51

Mesfin, F. B., & Al-Dhahir, M. A. (2023). Gliomas. *StatPearls*. https://www.ncbi.nlm.nih.gov/books/NBK441874/

Mohi ud din, A., & Qureshi, S. (2022). A review of challenges and solutions in the design and implementation of deep graph neural networks. *International Journal of Computers and Applications*, *45*(3), 221–230. https://doi.org/10.1080/1206212X.2022.2133805

Musin, O. R. (n.d.). *PROPERTIES OF THE DELAUNAY TRIANGULATION*.

Nair, A., Arvidsson, H., Gatica V, J. E., Tudzarovski, N., Meinke, K., & Sugars, R. V. (2022). A graph neural network framework for mapping histological topology in oral mucosal tissue. *BMC Bioinformatics*, *23*(1). https://doi.org/10.1186/S12859-022-05063-5

Nath, S. (2022). Network representation and analysis of energy coupling mechanisms in cellular metabolism by a graph-theoretical approach. *Theory in Biosciences*, *141*(3), 249–260. https://doi.org/10.1007/S12064-022-00370-0/FIGURES/6

Nickolls, J., Buck, I., Garland, M., & Skadron, K. (2008). Scalable Parallel Programming with CUDA. *Queue*, *6*(2), 40–53. https://doi.org/10.1145/1365490.1365500

Pantanowitz, L., Farahani, N., & Parwani, A. (2015). whole slide imaging in pathology: advantages, limitations, and emerging perspectives. *Pathology and Laboratory Medicine International*, 23. https://doi.org/10.2147/PLMI.S59826

Rathinabai, G. P., & Jeyakumar, G. (2021). Cell graph analysis through various construction methods. *Malaya Journal of Matematik*, *S*(1), 586–588. https://doi.org/10.26637/MJMS2101/0133

Sanchez-Lengeling, B., Reif, E., Pearce, A., & Wiltschko, A. B. (2021). A Gentle Introduction to Graph Neural Networks. *Distill*, *6*(9), e33. https://doi.org/10.23915/DISTILL.00033

Satya Mallick, & Krutika Bapat. (2018, December 10). *Shape Matching using Hu Moments*. https://learnopencv.com/shape-matching-using-hu-moments-c-python/

Shetty, J. K., Prasad, K. H., & Raghothaman, A. (2022). Challenges in the Histopathologic Diagnosis of Brain Tumors: An Institutional Experience in a Series of Cases. *J Health Allied Sci NU*, *12*, 412–416. https://doi.org/10.1055/s-0042-1742372

Slurm Workload Manager - Overview. (2021, August 6). https://slurm.schedmd.com/overview.html

Sudbø, J., Bankfalvi, A., Bryne, M., Marcelpoil, R., Boysen, M., Piffko, J., Hemmer, J., Kraft, K., & Reith, A. (2000). Prognostic value of graph theory-based tissue architecture analysis in carcinomas of the tongue. *Laboratory Investigation*, *80*(12), 1881–1889. https://doi.org/10.1038/LABINVEST.3780198

Veličković, P., Casanova, A., Liò, P., Cucurull, G., Romero, A., & Bengio, Y. (2017). Graph Attention Networks. *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*. https://doi.org/10.1007/978-3-031-01587-8_7

Wang, Y., Wang, Y. G., Hu, C., Li, M., Fan, Y., Otter, N., Sam, I., Gou, H., Hu, Y., Kwok, T., Zalcberg, J., Boussioutas, A., Daly, R. J., Montúfar, G., Liò, P., Xu, D., Webb, G. I., & Song, J. (2022). Cell graph neural networks enable the precise prediction of patient survival in gastric cancer. *Npj Precision Oncology 2022 6:1*, *6*(1), 1–12. https://doi.org/10.1038/s41698-022-00285-5

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., & Yu, P. S. (2021). A Comprehensive Survey on Graph Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems*, *32*(1), 4–24. https://doi.org/10.1109/TNNLS.2020.2978386

Xu, J., Li, Z., Du, B., Zhang, M., & Liu, J. (2020). Reluplex made more practical: Leaky ReLU. *Proceedings - IEEE Symposium on Computers and Communications*, *2020-July*. https://doi.org/10.1109/ISCC50000.2020.9219587

Xu, K., Jegelka, S., Hu, W., & Leskovec, J. (2018). How Powerful are Graph Neural Networks? *7th International Conference on Learning Representations, ICLR 2019*. https://arxiv.org/abs/1810.00826v3

Yang, M., Meng, Z., & King, I. (2020). FeatureNorm: L2 feature normalization for dynamic graph embedding. *Proceedings - IEEE International*

Conference on Data Mining, ICDM, *2020-November*, 731–740. https://doi.org/10.1109/ICDM50108.2020.00082

You, J., Ying, R., & Leskovec, J. (2020). Design Space for Graph Neural Networks. *Advances in Neural Information Processing Systems*, *2020-December*. https://arxiv.org/abs/2011.08843v2

Zemlyachenko, V. N., Korneenko, N. M., & Tyshkevich, R. I. (1985). Graph isomorphism problem. *Journal of Soviet Mathematics*, *29*(4), 1426–1481. https://doi.org/10.1007/BF02104746

Zheng, Y., Gindra, R. H., Green, E. J., Burks, E. J., Betke, M., Beane, J. E., & Kolachalama, V. B. (2022). A Graph-Transformer for Whole Slide Image Classification. *IEEE Transactions on Medical Imaging*, *41*(11), 3003–3015. https://doi.org/10.1109/TMI.2022.3176598

A APPENDICES

A.1 Model Performances

Dataset	*Accuracy*	*Precision*	*Recall*	*F1 Score*	*AUC*
Delaunay	0.942	0.964	0.945	0.953	0.994
KNN (K=10)	0.951	0.953	0.968	0.958	0.998
KNN (K=15)	0.942	0.947	0.967	0.955	0.994
KNN5 (K=5)	0.946	0.953	0.965	0.957	0.997

Table 6. Average performance metrics by Dataset. Higher is better.

Layer Type	*Accuracy*	*Precision*	*Recall*	*F1 Score*	*AUC*
GAT	0.867	0.868	0.969	0.907	0.975
GCN	0.852	0.856	0.967	0.898	0.971
GIN	0.883	0.897	0.953	0.915	0.976
GraphSAGE	0.853	0.854	0.969	0.897	0.972

Table 7. Average performance metrics by Layer Type. Higher is better.

Number of Layers	*Accuracy*	*Precision*	*Recall*	*F1 Score*	*AUC*
Input	-0.090	-0.101	0.009	-0.094	-0.124
Message Passing	-0.017	-0.004	-0.082	-0.029	-0.052
Task-Based	0.015	0.025	-0.063	0.006	0.012

Table 8. Correlation of number of layers with performance metrics. Higher is better.

Residual connection	*Accuracy*	*Precision*	*Recall*	*F1 Score*	*AUC*
skipconcat	0.883	0.887	0.972	0.919	0.982
skipsum	0.887	0.893	0.964	0.919	0.981
stack	0.733	0.739	0.939	0.811	0.921

Table 9. Average performance metrics by Residual Connection type. Higher is better.

Dropout Rate	*Accuracy*	*Precision*	*Recall*	*F1 Score*	*AUC*
0.0	0.954	0.973	0.953	0.962	0.996
0.1	0.933	0.929	0.970	0.947	0.994

Table 10. Correlation of Dropout Rate with performance metrics. Higher is better.

A.2 Model Complexities and Training Times

Number of Layers	*Avg. num. of params.*	*Avg. time per epoch + std.*
Input	-0.007	-0.023 ± -0.035
Message Passing	0.312	0.066 ± 0.042
Task-Based	0.221	-0.029 ± 0.036

Table 11. Correlation Matrix of number of layers with number of parameters and training time of the models. Lower is better.

Layer Type	*Num. of params.*	*Avg. time per epoch + std.*
GAT	29195.667	0.011 ± 0.001
GCN	28321.000	0.009 ± 0.000
GraphSAGE	30987.667	0.007 ± 0.000
GIN	32127.525	0.007 ± 0.000

Table 12. Average number of parameters and training time by Layer Type. Lower is better.

Residual connection	*Avg. num. of params.*	*Avg. time per epoch + std.*
skipconcat	62786.616	0.009 ± 0.001
skipsum	4937.048	0.008 ± 0.000
stack	4929.000	0.007 ± 0.000

Table 13. Average number of Parameters and Training Time by Residual Connection. Lower is better.