EPFL

Thèse n° 9447

Graph Representation Learning in Computational Pathology

Présentée le 21 février 2022

Faculté des sciences et techniques de l'ingénieur
Laboratoire de traitement des signaux 5
Programme doctoral en génie électrique

pour l'obtention du grade de Docteur ès Sciences

par

Guillaume JAUME

Acceptée sur proposition du jury

Prof. P. Vandergheynst, président du jury
Prof. J.-Ph. Thiran, Dr M. Gabrani, directeurs de thèse
Prof. H. Müller, rapporteur
Dr P. Moulin, rapporteur
Prof. P. Frossard, rapporteur

École polytechnique fédérale de Lausanne

2022

In anything at all, perfection is finally attained not
when there is no longer anything to add,
but when there is no longer anything to take way,
when a body has been stripped down to its nakedness.
— Antoine de Saint-Exupéry

To my parents,

Acknowledgements

After four years, the completion of this dissertation marks the end of my PhD. This journey, sometimes exhilarating, sometimes filled with doubts, has profoundly changed me. And if this period has challenged me in ways I could not have imagined, it has above all been incredibly enriching from a personal and professional point of view. This adventure would not have been possible without the support of many people, which I will humbly try to thank.

First of all, I would like to thank my EPFL supervisor, Prof. Jean-Philippe Thiran for his guidance throughout my PhD. Thanks for your trust and support that allowed me exploring different research directions and topics.

Many thanks to Dr. Maria Gabrani, my PhD co-supervisor at IBM, for her unwavering dedication in supporting my research and work. Of course, none of this adventure would have been possible without you, thanks for everything you have done for me.

I would also like to acknowledge the members of my thesis jury: Prof. Pierre Vandergheynst, the president of the jury, Prof. Pascal Frossard, Dr. Pierre Moulin, and Prof. Henning Müller, for their insightful comments and questions during the defense.

I would like to thank all my co-authors, with whom I had the chance to work. Pushpak, we have co-signed almost all our PhD publications together. Behind this are hundreds of meetings, calls, brainstorming sessions. You were always coming up with new ideas, always trying to go further, constantly questioning our work – I am extremely grateful for all our discussions, which have always inspired me. But our relationship goes beyond work, thanks for all the moments we spent playing pool together, for sharing your love of Indian food, and of course I can only be impressed by your dedication to increasing my tolerance for spicy food. Anphi, yes, we co-authored a publication together, but what I will remember most is that incredible trip to New Orleans and Miami - it has been an extraordinary adventure, that has been the starting point of a true friendship. Behzad, thanks for your advice and support during the four years of my PhD, your constant optimism and determination helped me believe in our work. Antonio, I remember that you were one of the first to encourage me to pursue a PhD when I was doing my master's thesis – I consider that decision one of the best of my life, and I want to thank you for pushing me in that direction. Working at the intersection of two fields, deep learning and pathology, is also a great opportunity to foster collaborations and learn about new domains. Throughout my PhD, I was lucky to work with amazing medical doctors, biologists and pathologists. Dr. Maryse Fiche, you never hesitated to take your free time to help me strengthen my modest, to say the least, knowledge of pathology, I owe you a great deal of thanks. Finally, a warm thank you to all the students I had the chance to supervise

during my thesis, Maria Halushko, Atul Kumar, Lauren Alisha Fernandez, Valentin Anklin, I hope you were able to learn as much as I learned from you.
The (almost) five years that I spent in IBM Research would not have been the same without all my outstanding C-HCLS colleagues. I would like to thank Anca, with whom I shared the office for two years, for all the amazing discussions (and chocolate croissant), Kevin T. for never forgetting to send me the latest news about a big transfer to Barca or PSG, Sasha (Kim) and Sonali, for always bringing your *joie-de-vivre*. Of course, there is more at IBM than the C-HCLS group, and I would like to thank Pauline and Kevin P., with whom the coffee breaks were a breath of fresh air during the long working days. I am also grateful to Gabriel, who accompanied me throughout my studies at EPFL and IBM Research, first by offering me a semester project in the LTS5, then by putting me in relation with Maria, and finally by mentoring the beginning of my PhD. Your advice and trust have sincerely helped me, and I would not be where I am today without you.
As the saying goes, *work, love and play are the great balance wheels of man's being*, and I think I have found this balance in Zurich. My life would not have been the same without all the amazing people around me. I would like to thank Malo, for being such a great riding/running partner, Cyril, for organizing amazing rooftop parties and for attempting to teach me the fundamentals of the rhythmic, Gaspard, for never forgetting to send me cat pictures, Pauline, for always organizing amazing weekend trips, Kevin P., for all those games of badminton and rackets at the lake, Manon, for bringing a little taste of the South of France to Zurich, Hugo for being (almost) as crazy as me with the Derrière le Miroir, Pol for all these discussions where we shared our love for Renoir, Charlotte for cooking the best homemade bread and cinnamon rolls in town, Hector for the Sunday night dinners at my place and the Catan games, Andrew, for teaching American slang, Alice, for the countless dinners at your place, Louis for the amazing trip to New York, Ahmed for the parties at Frieda's. Thank you for the countless memories that will remain forever in my mind. A special thanks to my flatmate, David, who has (so far) managed to put up with me, thanks for the pasta dishes, pancakes. Last, but certainly not least, thanks Tanja for your endless support over the past four years. You have been involved more than anyone else, and have always been there to listen and find the words in moments of doubt. This thesis would not have been the same if you had not been present.
Finally, I want to thank my family, without whom I would not be here. Juliette, thank you for the interest you have always shown in my thesis, it has helped me a lot to improve my communication and vulgarization skills. *Papi*, thanks for sparking my curiosity as a child, my interest in science grew in large part because of you, and I will be forever appreciative. *Papa et Maman*, I am grateful for all that you have given me, without expecting anything in return, and I consider myself lucky to have been able to receive such support. You have always been able to find the balance, which can sometimes be fragile, between reminding us of the importance of education and giving us the freedom to grow outside the classroom. In all humility, look at this thesis as the materialization of your success as parents in providing a quality education to your children, something that you have achieved with flying colors for both Juliette and me.

Zurich, January 25, 2022 G. J.

Abstract

Advances in scanning systems have enabled the digitization of pathology slides into Whole-Slide Images (WSIs), opening up opportunities to develop Computational Pathology (CompPath) methods for computer-aided cancer diagnosis and prognosis. CompPath has been primarily developed using models based on Convolutional Neural Networks (CNNs), building on the recent successes in Computer Vision. A series of promising approaches have been proposed for nuclei segmentation and classification, tumor detection, tumor grading, among others. However, CNN-based methods suffer from several limitations. First, it is challenging to model both fine-grained nuclei-level information and long-range inter-glandular dependencies. Second, there is a discrepancy between the pixel-based analysis of CNNs and the histological entity-centered analysis employed for pathological diagnosis, which in turn can hinder model transparency. Third, the inherent complexity of training networks on large histology images with limited annotations constrains its learning capabilities.

Instead, we propose an analytical paradigm shift, where we view and analyze histology images as a *set of biological entities interacting with each other*. Specifically, an image is represented as an *entity-graph* where nodes depict biological entities and edges encode interactions between these entities. *Entity-graphs* are further processed by a Graph Neural Network (GNN) model to jointly encode the entity morphological attributes and topological distribution, towards tissue phenotyping. In this thesis, we study three research directions in CompPath, namely, scalability, interpretability and explainability, and weakly-supervised learning.

First, histology images are orders of magnitude larger than natural images, where diagnostically relevant regions can represent only a fraction of the image. We propose a *scalable* hierarchical cell-to-tissue representation (HACT) and GNN model, HACT-Net, for learning on arbitrary large inputs. We show the capabilities of HACT-Net on our proposed BRACS dataset, the largest cohort to date for breast tumor Regions-of-Interest subtyping.

Second, computer-aided diagnostic tools must be transparent and their decision-making process justified. By shifting the analysis from pixel- to entity-based, we make the input space interpretable for pathologists that can better relate to the model input. We further propose entity-centric graph explainers, exemplified with *cell-graph* model explainability, along with novel metrics to evaluate explanations based on entity-level pathological concepts.

Third, acquiring ground-truth data to train deep learning systems requires pathologists to provide specific annotations, which is time-consuming, expensive, and subject to inter- and intra-observer variability. We therefore propose WHOLESIGHT, a method that reduces annotation requirements to WSI-level labels only, for joint *classification* and *segmentation* of

WSIs. We show the capabilities of WHOLESIGHT for Gleason pattern segmentation and grading on multi-source WSI prostate datasets. The generalization properties of WHOLESIGHT are further evaluated on unseen cohorts, and compared to Bayesian variants to strengthen the estimation of the model uncertainty. Finally, we introduce HISTOCARTOGRAPHY, a novel python library designed to accelerate the development of graph analytics in CompPath.

Keywords: computational pathology, graph neural network, whole-slide image classification, gleason grading, graph representation learning, deep learning, cancer grading, cancer subtyping.

Résumé

Les progrès réalisés dans les systèmes de numérisation permettent désormais de convertir les lames de pathologie en images (WSIs), ouvrant la voie au développement de méthodes de pathologie computationnelle (CompPath) pour le diagnostic du cancer assisté par ordinateur. Jusqu'à présent, la CompPath a été développé à l'aide de modèles basés sur les réseaux neuronaux convolutifs (CNNs), en capitalisant sur les récents succès de la vision par ordinateur. Une série d'approches prometteuses ont par exemple été proposées pour la segmentation et la classification des noyaux cellulaires, ou la détection et la classification de tumeurs. Cependant, les méthodes basées sur les CNNs présentent plusieurs limitations. Premièrement, il est difficile de modéliser à la fois des détails de l'image et son contexte. Deuxièmement, il existe une inadéquation entre l'analyse basée sur les pixels des CNNs et l'analyse centrée sur les entités histologiques utilisée pour le diagnostic pathologique, ce qui peut affecter la transparence du modèle. Troisièmement, la complexité inhérente à l'entraînement des réseaux sur de grandes images histologiques avec des annotations peu détaillées limite leurs capacités d'apprentissage.

A contrario, nous proposons un changement de paradigme analytique, où nous représentons les images histologiques comme un ensemble d'entités biologiques interagissant les unes avec les autres. Un graphe d'entités est créé où les noeuds représentent des entités biologiques et les arêtes symbolisent les interactions entre ces entités. Les graphes d'entités sont ensuite analysés par un réseau de neurones de graphe (GNN) afin d'encoder les attributs morphologiques des entités et leur distribution topologique. Plus précisément, dans cette thèse, nous nous penchons sur trois axes de recherche en CompPath, à savoir la scalabilité, l'interprétabilité et l'explicabilité, ainsi que l'apprentissage faiblement supervisé.

Premièrement, les images histologiques sont nettement plus grandes que les images ordinairement utilisées en vision par ordinateur. Nous proposons une représentation hiérarchique et scalable des tissus pour l'apprentissage sur des données de grande taille. Nous démontrons ses capacités sur les données BRACS, une cohorte d'images pour le sous-typage de tissus mammaires.

Deuxièmement, les outils de diagnostic assistés par ordinateur doivent être transparents. En passant d'une analyse basée sur les pixels à une analyse basée sur les entités, nous rendons l'espace d'entrée interprétable pour les pathologistes. Nous proposons en outre des explicateurs de graphes, illustrés par l'explicabilité des modèles de graphes cellulaires, ainsi que de nouvelles métriques.

Troisièmement, l'acquisition d'annotations pour entraîner les systèmes d'apprentissage est

longue, coûteuse et sujet à la variabilité inter- et intra-observateur. Nous proposons une méthode qui réduit les exigences d'annotation aux seules mentions du type de la WSI, en vue de sa classification et segmentation. Nous démontrons ses capacités sur des ensembles de données prostatiques pour la prediction du grade de Gleason. Les propriétés de généralisation du modèle sont ensuite évaluées sur des cohortes externes, et comparées à des variantes bayésiennes pour renforcer les estimations d'incertitude. Enfin, nous présentons Histocartography, une nouvelle bibliothèque python conçue pour accélérer le développement des graphes en CompPath.

Mot-clés : pathologie computationnelle, réseau neuronal de graphe, classification d'images de lames entières, classification de gleason, apprentissage de la représentation de graphes, apprentissage profond, gradation du cancer, sous-typage du cancer.

Contents

List of Figures

List of Tables

Acronyms

AI Artificial Intelligence. 1, 2, 11, 35, 131, 141

API Application Programming Interface. 148, 149, 152

CAD Computer-Aided Diagnosis. 52, 107, 109, 111, 112

CAM Class Activation Map. 40, 143, 144

CNN Convolutional Neural Network. 2, 3, 14–16, 29, 35, 40, 54, 58–60, 63, 68–76, 93, 110, 124, 139, 143, 150, 151

CompPath Computational Pathology. 1, 3–5, 12, 49, 52, 54, 88, 104, 107, 110–112, 142, 147–149

CPU Central Processing Unit. 78, 121, 149, 154

CV Computer Vision. 1, 11, 36

DGL Deep Graph Library. 70, 98, 121, 152, 153

DigPath Digital Pathology. 1, 4, 49

DL Deep Learning. 1, 3–5, 11, 13, 35, 54, 87–90, 104, 107, 108, 111, 119, 142, 147, 148, 151, 152

ECE Expected Calibration Error. 123, 130

EM Expectation Maximization. 110

FDA Federal Drug Administration. 52

GAP Global Average Pooling. 143

GCN Graph Convolutional Network. 18, 19

GIN Graph Isomorphism Network. 25, 26, 29, 30, 33, 71, 113, 115, 116, 121, 139, 152

GNN Graph Neural Network. 3, 4, 11, 15–19, 21, 22, 25, 26, 28, 30–33, 35, 36, 39–43, 54, 59–61, 65, 68–76, 79, 88–93, 98, 104, 112–118, 121, 123, 124, 131, 139–141, 143, 148, 152, 153

GPU Graphics Processing Unit. 70, 78, 98, 121, 149, 154

H&E Hematoxylin & Eosin. 51, 53, 61, 66, 112, 114, 139, 142, 149

k-NN k-Nearest Neighbors. 54, 63, 91, 140, 149, 151, 157

MIL Multiple Instance Learning. 60, 108, 110, 124

ML Machine Learning. 54, 148, 150

MLE Maximum Likelihood Estimation. 114, 119

MLP Multi-layer Perceptron. 14, 16, 29, 65, 68–70, 92, 113, 116, 118, 121, 122, 125

MPNN Message Passing Neural Network. 4, 11, 15–19, 21, 22

NLL Negative Log-Likelihood. 114, 123, 129

NLP Natural Language Processing. 11, 36

PNA Principal Neighborhood Aggregation. 29, 30, 33, 65, 69–73, 139, 152

RAG Region Adjacency Graph. 64, 115, 151

ReLU Rectified Linear Unit. 16, 31, 32, 40, 121, 145

RNN Recurrent Neural Network. 13, 35, 60, 110

RoI Region-of-Interest. 78, 139, 154, 156

SGD Stochastic Gradient Descent. 114

SLIC Simple Linear Iterative Clustering. 63, 115, 141, 150, 155

SVD Singular Value Decomposition. 53

t-SNE t-distributed stochastic neighbor. 130, 131

TMA Tissue-Micro Array. 2

TRoI Tumor Region-of-Interest. 2, 5, 56, 59–61, 63–70, 73, 75, 77–79, 91, 92, 97–101, 103, 155, 158

UI User Interface. 53, 148

VAE Variational Auto-Encoder. 110

WSI Whole-Slide Image. 1, 2, 5, 52, 59, 60, 66, 77, 78, 108–112, 115, 116, 118–124, 126, 127, 130, 131, 140–142, 148, 149, 151, 154, 157, 158

XAI Explainable AI. 5, 36

Introduction

Motivation

Advances in scanning systems have enabled the digitization of pathology slides into high-resolution Whole-Slide Images (WSIs), heralding the era of Digital Pathology (DigPath) (Mukhopadhyay et al., 2017). In parallel, Deep Learning (DL) has revolutionized Computer Vision (CV), with the development of algorithms capable of detecting, classifying and, segmenting images with unprecedented accuracy (Deng et al., 2009a; Krizhevsky et al., 2012; He et al., 2016). These breakthroughs are at the root of Computational Pathology (CompPath), which paves the way for the creation of Artificial Intelligence (AI)-powered tools for objective cancer diagnosis and prognosis, as well as for the prediction of treatment response and resistance.

The stakes are high. Almost four in ten Americans will be diagnosed with cancer in their lifetime. Prostate cancer, the second most frequently diagnosed cancer in men in the U.S., has registered 250,000 new cases in 2021 and is responsible for 35,000 deaths. Over the same period, 280,000 new cases of invasive breast cancer were diagnosed in the U.S., causing 43,000 deaths, the highest number of deaths among women with cancer (Duggan et al., 2021). In addition, the overall incidence rate of cancer cases per year is increasing. In the U.S., the incidence rate of breast cancer is steadily increasing by 0.5% per year (Siegel et al., 2020) to the figure of approximately 1 in 8 women who will develop invasive breast cancer in their lifetime. Moreover, the number of pathologists, who play a central role in diagnosing and treating cancer patients, is gradually decreasing. In the U.S., a decrease of 18% has been observed between 2007 and 2017, resulting in a 42% increase in average workload (Wilson et al., 2018). Additionally, the practice of pathology is subject to its own challenges. In particular, even though diagnostic criteria for cancer are established (Tan et al., 2019), the continuous nature of histologic features phenotyped across the diagnostic spectrum leaves room for inconsistencies, with significant intra- and inter-observer variability (Gomes et al., 2014; Elmore et al., 2015). Moreover, manual slide inspection is a tedious and time-consuming process, which would benefit from automation and standardization, thus reducing the workload for pathologists. The aforementioned factors are therefore motivating the development of computer-aided diagnosis tools (Bulten et al., 2021; Campanella et al., 2019).

Advances in DL have already enabled the development of clinically-relevant pathology tasks (Ibrahim et al., 2020), such as nuclei segmentation (Kumar et al., 2017; Graham et al., 2019a), nuclei

classification (Verma et al., 2021), gland segmentation (Graham et al., 2019b; Binder et al., 2019), tissue segmentation (Mehta et al., 2018; Mercan et al., 2019b), tumor detection (Aresta et al., 2019; Bejnordi et al., 2019; Pati et al., 2018), WSI tumor grading (Lu et al., 2020; Shaban et al., 2020; Tellez et al., 2019a), and tumor staging (Aresta et al., 2019; Mercan et al., 2019a). Recent work even showed that AI-assisted diagnosis could yield better cancer grading than that performed by pathologists alone (Bulten et al., 2021; Campanella et al., 2019).

These advances are based on the advent of Convolutional Neural Networks (CNNs), which have been originally developed to operate on natural images. Histology images, however, possess unique features that make them challenging to be modeled. First, histology images are large. WSIs, obtained by digitizing a tissue specimen, can be as large as $100'000 \times 100'000$ pixels (see Figure 1). Even Tissue-Micro Arrays (TMAs) and Tumor Regions-of-Interest (TRoIs) remain orders of magnitude larger than ImageNet images (Deng et al., 2009a). Second, a tumor region represents a *hierarchical* composition, where nuclei will form tissue regions that further form large glandular structures. Consequently, the diagnosis should be based on a multi-scale analysis, where fine-grained information should be related to coarser high-level patterns. It also means that, in cases where large non-informative benign regions are present, only a fraction of the image is relevant for diagnosis. To address the aforementioned challenges, most previous CNN-based methods adopt a patch-based processing approach. Specifically, a (large) input image is tiled into small fixed-size patches, that are individually processed by a pre-trained CNN. Then, an aggregation function pools the information from each patch to build an image-level embedding used for downstream tasks, *e.g.*, tumor grading.

While being applicable, these approaches suffer from several limitations. First, there is a trade-off between operation resolution and how much context is included in a patch. In other words, context and resolution cannot be optimally leveraged in this setting. Second, the aggregation operation would typically discard the *relational* information between the patches by treating patch embeddings as independent units. Third, patch-level processing cannot efficiently combine multi-scale information, making it hard to incorporate tissue compositionality in the model. Fourth, pixel-based processing is wholly detached from biological reasoning, making these approaches hard to interpret.

Instead, the core idea of this thesis is a paradigm shift, where histology images are analyzed as an *interacting set of biological entities*. Specifically, we propose to use *graph* representations, as a means to encode the tissue-to-function relationship, where nodes represent biological entities, and edges represent entity-entity interactions. The nodes, *i.e.*, the entities, can be encoded at the appropriate scale, depending on the task at hand. The edges act as support to efficiently propagate information from one entity to another and highlight higher-level patterns relevant for modeling tissue compositionality. Also, as the nodes are biologically defined, pathologists can make the link with their own understanding of the problem. Thus, this representation allows better adequacy with the way pathologists reason and enables more transparency in the algorithm's operation.

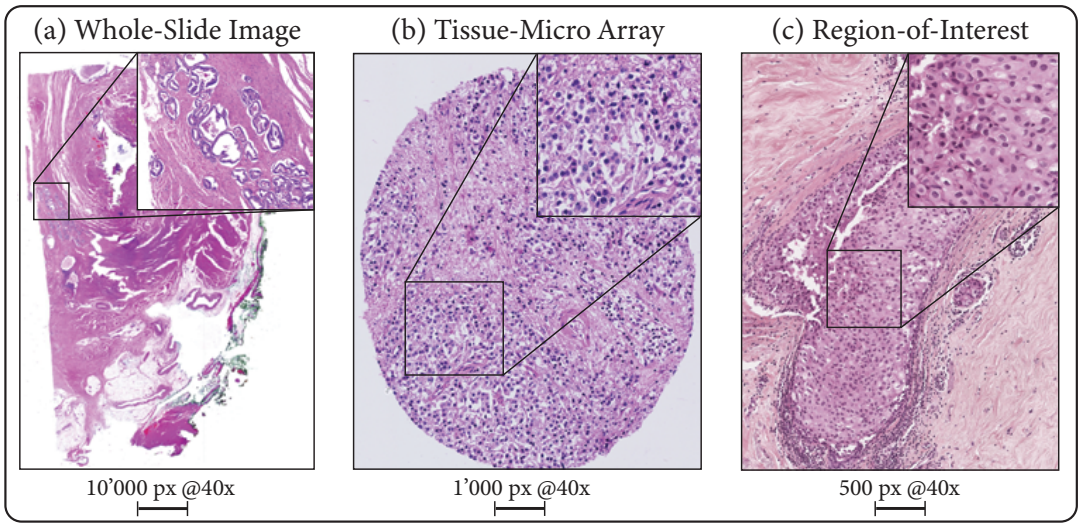


Figure 1 – Examples of histology images. (a) a tissue resection prostate WSI, (b) a TMA obtained from a core needle biopsy of prostate tissue, (c) a TRoI extracted from a breast biopsy.

We build on recent advances in graph representation learning and the development of Graph Neural Networks (GNNs). Today, GNNs are an integral part of the DL toolbox and have been successfully applied to various applications, *e.g.*, for antibiotic discovery (Stokes et al., 2020) or prediction of polypharmacy side effects (Zitnik et al., 2018). The flexibility of GNNs enables the modeling of complex histological structures, with an arbitrarily large number of biological entities and interactions, as well as entity descriptors.

In this thesis, we show that entity-centric graph representations of histology images combined with GNNs present significant advantages over traditional CNN-based methods. Specifically, analyzing histology images through the lens of graphs allows us to explore three research directions in CompPath, namely: (i) *scalability* to build models that can operate on images of arbitrary size and shape, (ii) *interpretability*, to devise transparent and explainable methods, and (iii) *weakly-supervised* training settings to work with limited annotations, which are often expensive and time consuming to acquire.

Thesis outline

This thesis is outlined as follows:

Part I: The first part of the dissertation covers graph representation learning and GNNs. It lays down the theoretical foundations to learn and understand graph models in different training scenarios.

Chapter 1: This chapter introduces graphs and GNNs, along with necessary notation, defini-

tions, and graph theory to start working with graph models confidently. An overview of the main graph learning tasks is provided. Then, by presenting a set of graph models *desiderata*, we introduce and justify the Message Passing Neural Network (MPNN) framework, which lies at the heart of most GNN models. We also present a spectral view of GNNs, where we highlight that some formulations of GNN layers amount to a convolution operation generalized to graph-structured data.

Chapter 2: One of the main questions in graph representation learning is to find the GNN architecture that would lead to the best graph model. An approach is to study the expressive power of the resulting GNN, with the end goal of being as expressive as graph isomorphism, *i.e.*, having a GNN model that *can* infer an injective mapping between the graph space and the embedding space. This chapter builds on recent work that established that some instances of MPNNs can be as powerful as the Weisfeiler-Lehman test of isomorphism for distinguishing node-attributed graphs. We generalize these results to *directed*, *node*- and *edge*-attributed graphs, and show that the resulting provably powerful method, EDGNN, competes with state-of-the-art methods.

Chapter 3: The ability to interpret DL predictions is crucial for real-world deployment, especially in high-risk settings such as computer-aided diagnosis. This chapter introduces a setting based on node-centric post-hoc interpretability to explain GNN models. Specifically, four graph explainers are introduced, based on gradient importance (GRAPHGRAD-CAM and GRAPHGRAD-CAM++), node pruning (GNNEXPLAINER), and layer-wise relevance propagation (GRAPHLRP). The proposed graph explainers provide node-level scores that characterize their importance towards the prediction of a given graph. Complementary mathematical derivations are included in Appendix A.

Part II: The second part of this thesis focuses on the analysis, understanding, and classification of histology images using *entity-graphs*. Graph representations and graph learning algorithms are employed to address three central challenges in computed-aided pathological diagnosis. Namely, (i) scaling to giga-pixel images without the need for tile-based representations, (ii) interpretable and explainable models to build trust between all stakeholders involved, *i.e.*, the medical staff, the patients and the algorithm, and (iii) weakly-supervised settings for learning with limited annotations. The core contributions of this thesis are included in this part.

Chapter 4: This chapter introduces preliminaries about pathology, DigPath and CompPath. An overview of the pathological diagnosis procedure is provided to further motivate the digitization and automation of pathology in clinical routine. An introduction to basic DigPath tools and image processing used in most CompPath projects, *e.g.*, stain normalization, is also provided. Finally, this chapter provides a high-level overview of the current state-of-the-art in graph-based histological image analysis, with methods ranging from cell-graph modeling, to hierarchical representations and patch-graph processing.

Chapter 5: This chapter presents HACT and HACT-Net, a HierArchical Cell-to-Tissue repre-

sentation and model, respectively, for learning on histology images. HACT includes a low-level cell-based representation of the image, combined with a high-level tissue-based representation and cell-to-tissue hierarchy to efficiently encode the tissue morphology. This chapter also introduces BRACS, the largest to date, dataset for breast carcinoma subtyping of TRoIs. In the proposed study, HACT-Net outperforms state-of-the-art DL approaches. HACT-Net even proves to outperform pathologists, especially in challenging atypical TRoI classification. This work marks an important step towards the use of DL algorithms for clinical diagnostic purposes.

Chapter 6: This chapter emphasizes the need for *explainable* and *interpretable* models, which provide sample-level explanations that are intuitive to pathologists. Specifically, a series of entity-graph explainers are proposed that produce sparse and accurate explanations in the node-space, *i.e.*, at nuclei-level. By shifting the analysis from pixels to nuclei, prior pathological knowledge can be leveraged to better understand what the algorithm is focusing on. This study concludes that graph explainers highlight important morphological, *e.g.*, nuclei chromatin, and topological concepts, *e.g.*, nuclei density, in agreement with pathologists' diagnostic criteria. This work is the first of its kind to prove the relevance of entity-graph representations for Explainable AI (XAI). Complementary qualitative experiments are provided in Appendix C.

Chapter 7: Segmenting histopathology images into diagnostically relevant regions is imperative to support timely and reliable decisions by pathologists. In this chapter, we propose WHOLESIGHT, a weakly-supervised semantic segmentation method using tissue-graphs, to jointly segment and classify whole-slide histopathology images of arbitrary shape and size. WHOLESIGHT first constructs a tissue-graph representation for an input image, where the nodes depict tissue regions, and the edges describe interactions among tissue regions. Subsequently, the method employs a *graph-classification head* to classify WSIs, followed by a post-hoc feature attribution technique to derive node-level pseudo labels. Finally, a *node classification head* is trained using the pseudo labels to segment WSIs. WHOLESIGHT is evaluated on three public prostate cancer WSI datasets from three pathology labs, proving its classification and segmentation capabilities. Further, two Bayesian variants, WHOLESIGHT-MCD and WHOLESIGHT-DE, are proposed based on MC-dropout and deep ensembles, which improve the generalization of WHOLESIGHT over external test datasets. The generalization capabilities of the methods are quantified in terms of segmentation and classification performance, uncertainty estimations, and model calibration analyses. Complementary experiments are provided in Appendix D. In addition, Appendix B presents HistoCartography, a library designed to ease the development of graph-based CompPath tools, and used to implement the pipelines described in Chapter 5,6,7.

Chapter 8: This chapter summarizes the main contributions of the thesis by discussing the findings and their limitations. Furthermore, a perspective on future challenges and opportunities is provided, concluding the thesis.

Contributions

The main contributions of this thesis are summarized as (in order of appearance):

- EDGNN, a novel GNN for learning on the directed node- and edge-labeled graphs;
- An expressivity analysis of EDGNN, concluding that EDGNN can be as expressive as the *1*-dimensional Weisfeiler–Lehman test of graph isomorphism;
- A set of four architecture-agnostic graph explainers that produce node-level importance scores for GNN interpretability;
- HACT and HACT-Net, a novel representation and neural network, respectively, for learning on histology images with entity-based processing;
- BRACS, the largest to date dataset of breast carcinoma tumor Regions-of-Interest for tumor subtyping;
- A method to analyze graph explanations with interpretable entity-level concepts, exemplified with a nuclei-level analysis of BRACS samples;
- WHOLESIGHT, a novel pipeline for joint classification and segmentation of prostatic WSIs using weakly-supervised learning;
- A study of generalization, uncertain estimation and calibration of WHOLESIGHT and Bayesian variants on *in-domain* and *out-of-domain* cohorts;
- HistoCartography, a Python library that includes image processing tools, graph building helpers, graph models, and graph explainers unified under a user-friendly API.

Publications

Publications integrated in the thesis

- Chapter 2: "EDGNN: A Simple and Powerful GNN for Directed Labeled Graphs", **Guillaume Jaume***, An-phi Nguyen*, Maria Rodriguez Martinez, Jean-Philippe Thiran, Maria Gabrani. In *International Conference on Learning Representations (ICLR) workshop on Representation Learning on Graphs and Manifolds*, 2019 (Jaume et al., 2019).
- Chapter 3 and Chapter 6: "Quantifying Explainers of Graph Neural Networks in Computational Pathology", **Guillaume Jaume***, Pushpak Pati*, Behzad Bozorgtabar, Antonio Foncubierta, Anna Maria Anniciello, Florinda Feroce, Tilman Rau, Jean-Philippe Thiran, Maria Gabrani, Orcun Goksel. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021 (Jaume et al., 2021b).

- Chapter 3 and Chapter 6: "Towards Explainable Graph Representations in Digital Pathology", **Guillaume Jaume***, Pushpak Pati*, Antonio Foncubierta, Florinda Feroce, Giosue Scognamiglio, Anna Maria Anniciello, Jean-Philippe Thiran, Orcun Goksel, Maria Gabrani. In *International Conference on Machine Learning (ICML), ICML Workshop on Computational Biology*, 2020, **[Best Paper Award]** (Jaume et al., 2020).

- Chapter 5: "Hierarchical Graph Representations in Digital Pathology", Pushpak Pati*, **Guillaume Jaume***, Antonio Foncubierta, Florinda Feroce, Anna Maria Anniciello, Giosuè Scognamiglio, Nadia Brancati, Maryse Fiche, Estelle Dubruc, Daniel Riccio, Maurizio Di Bonito, Giuseppe De Pietro, Gerardo Botti, Jean-Philippe Thiran, Maria Frucci, Orcun Goksel, Maria Gabrani. In *Medical Image Analysis*, 2021 (Pati et al., 2021a).

- Chapter 5: "HACT-Net: A Hierarchical Cell-to-Tissue Graph Neural Network for Histopathological Image Classification", Pushpak Pati*, **Guillaume Jaume***, Lauren Alisha Fernandes, Antonio Foncubierta-Rodríguez, Florinda Feroce, Anna Maria Anniciello, Giosue Scognamiglio, Nadia Brancati, Daniel Riccio, Maurizio Di Bonito, Giuseppe De Pietro, Gerardo Botti, Orcun Goksel, Jean-Philippe Thiran, Maria Frucci, Maria Gabrani. In *International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI), MICCAI Workshop on Graphs in Medical Imaging*, 2020, **[Best Paper Award]** (Pati et al., 2020).

- Chapter 7: "Learning Whole-Slide Segmentation from Inexact and Incomplete Labels using Tissue Graphs", Valentin Anklin*, Pushpak Pati*, **Guillaume Jaume***, Behzad Bozorgtabar, Antonio Foncubierta-Rodríguez, Jean-Philippe Thiran, Mathilde Sibony, Maria Gabrani, Orcun Goksel. In *Medical Image Computing and Computer Assisted Interventions (MICCAI)*, 2021 (Anklin et al., 2021).

- Chapter 7: "Weakly Supervised Learning for Joint Whole-Slide Segmentation and Classification in Prostate Cancer", **Guillaume Jaume***, Pushpak Pati*, Behzad Bozorgtabar, Jean-Philippe Thiran, Orcun Goksel Maria Gabrani. In *Preprint*, 2021 (Jaume et al., 2021c).

- Appendix B: "HistoCartography: A Toolkit for Graph Analytics in Digital Pathology", Guillaume Jaume*, Pushpak Pati*, Valentin Anklin, Antonio Foncubierta, Maria Gabrani. In *International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI), Third MICCAI workshop on Computational Pathology*, 2021, **[Best Software Paper Award]**(Jaume et al., 2021a).

External publications

Several publications published during the thesis do not appear in this manuscript. For the sake of completeness, we provide the list below:

- "Image-Level Attentional Context Modeling Using Nested-Graph Neural Networks",

Guillaume Jaume, Behzad Bozorgtabar, Hazim Kemal Ekenel, Jean-Philippe Thiran, Maria Gabrani. In *Conference on Neural Information Processing Systems (NeurIPS), Workshop on Relational Representation Learning*, 2018.

- "FUNSD: A Dataset for Form Understanding in Noisy Scanned Documents", **Guillaume Jaume**, Hazim Kemal Ekenel, Jean-Philippe Thiran. In *International Conference on Document Analysis and Recognition (ICDAR)*, 2019.

- "BRACS: A Dataset for BReAst Carcinoma Subtyping in H&E Histology Images", Nadia Brancati, Anna Maria Anniciello, Pushpak Pati, Daniel Riccio, Giosuè Scognamiglio, **Guillaume Jaume**, Giuseppe De Pietro, Maurizio Di Bonito, Antonio Foncubierta, Gerardo Botti, Maria Gabrani, Florinda Feroce, Maria Frucci. In *Under review*, 2021.

Graph Representation Learning Part I

1 The Graph Neural Network Model

A graph is a mathematical structure that can represent entities, or *nodes*, and the way they interact with each other, in the form of *edges*. Graphs are ubiquitous data as they can represent any complex system. For instance, molecules can be represented as graphs, where atoms represent nodes that are connected to each other via chemical bonds that represent the edges. Social networks are composed of users that represent the nodes, and the user interactions represent the edges. Differently, physical systems can also be represented as graphs where objects, *i.e.*, the nodes, interact with each other through physical forces, *i.e.*, the edges.

Learning to model graphs is of high significance to encode, understand, and predict the behavior of complex systems. Such models can be used to predict molecular properties associated to chemical compounds, recommend new connections to a user in a social network, identify communities to control the spread of a virus in a population, etc.

The advent of deep learning has brought significant breakthroughs in AI research with the development of *neural network* models able to learn on large-scale datasets in CV (Krizhevsky et al., 2012; He et al., 2017; Ren et al., 2017), Natural Language Processing (NLP) (Kenton et al., 2017), among others. In parallel, several research groups started to *extend* these successful neural approaches to graph-structured data, leading to the development of Graph Neural Networks.

In this chapter, we start by formally introducing graphs and some of their properties. We present an overview of the tasks that can benefit from the development of neural network-based graph models. Then, GNNs and the MPNN framework are introduced as universal graph models that can be used to encode any graph dataset. Finally, we provide a theoretical motivation justifying the design of MPNNs using graph spectral theory. We emphasize that graph representation learning is an extensively discussed topic in the field of DL today, and that this chapter is by no means a review of all existing approaches. Rather, it is intended to provide sufficient background knowledge to understand the contributions of this thesis. The reader can refer to Wu et al. (2020); Zhou et al. (2021, 2019b); Hamilton et al. (2020) for a review of GNN taxonomy and graph representation learning.

1.1 Graphs: definition and notation

A graph G is a pair (V_G, E_G), where V_G is the set of nodes and E_G is the set of edges. The number of nodes and edges in G are denoted as $|V_G|$ and $|E_G|$, respectively. The directed edge $(u, v) \in E_G$ for $u, v \in V_G$ is an edge starting in u and ending in v. The graph edges can be represented by an adjacency matrix, denoted as $A \in \mathbb{R}^{(|V_G| \times |V_G|)}$, where $A_{u,v} = 1$, if $(u, v) \in E_G$. When there is no ambiguity, the node and edge sets will simply be denoted as V and E, respectively.

For each node $v \in V_G$, we define its neighborhood as $\mathcal{N}(v) := \{u \in V_G \mid (v, u) \in E_G \vee (u, v) \in E_G\}$. When we are dealing with directed graphs, we distinguish between incoming neighbors $\mathcal{N}^I(v) := \{u \in V_G \mid (u, v) \in E_G\}$ and outgoing neighbors $\mathcal{N}^O(v) := \{u \in V_G \mid (v, u) \in E_G\}$. Naturally, $\mathcal{N}(v) = \mathcal{N}^I(v) \cup \mathcal{N}^O(v)$. The cardinalities $|\mathcal{N}(v)|$, $|\mathcal{N}^I(v)|$ and $|\mathcal{N}^O(v)|$ of the neighborhoods are referred to as the *degree*, the *in-degree* and the *out-degree* of node v, respectively, and are denoted as d_v, d_v^I, d_v^O.

In this thesis, we are concerned with *attributed* graphs $G := (V_G, E_G, H)$ where each node is associated to attributes that *characterize* it. Node attributes are denoted as $H \in \mathbb{R}^{|V| \times d}$, or at the node-level as $H_{v,.} := h(v) \in \mathbb{R}^d$. In some cases, we will also consider graphs with *edge* attributes, $G := (V_G, E_G, H, H_E)$, where the edge attributes are denoted as $H_E \in \mathbb{R}^{|E| \times d_E}$, or $H_{(u,v),.} := h_E(u, v) \in \mathbb{R}_E^d$ at edge-level. We refer to *discrete* node- and edge-attributes as *labels*, *e.g.*, atoms in a molecule, company names in a knowledge graph, etc. Multi-dimensional *continuous* node- and edge-attributes are referred to as *features*. Note that the graph signal processing community usually employs the term *signal* to refer to attributes. *Processed* labels and features are referred to as node- and edge-*embeddings*.

In the literature, the terms graph and network are often used interchangeably. In order to avoid confusion with neural *networks*, and in agreement with the deep graph community, we will only use the term graph. Similarly, nodes and vertices are both accepted terminologies. In this work, we will only be using the term node for its intuitive character.

1.2 Main tasks to learn on graphs

In this section, we provide an overview of common graph machine learning tasks (see Figure 1.1).

Graph classification: This task is based on supervised learning where we are given a set of graphs associated to a label that needs to be inferred. Graph classification is analogous to supervised image classification, *e.g.*, MNIST, CIFAR and ImageNet classification. For instance, molecular property predictions are popular graph classification tasks *e.g.*, mutagenicity or carcinogenicity characterization of chemical compounds. In CompPath, graph classification can be used to predict the aggressiveness of tumor regions represented as graphs (see Chapter 5).

Node classification: This task is defined in a semi-supervised learning setting, where we are

given a large, partially annotated graph. Nodes and edges are typically associated to additional attributes. During training, known node labels are used to train a model, then during inference, unknown nodes from the *same* graph are predicted. This setting breaks the i.i.d assumption of DL as the nodes to classify are connected to each other, and therefore influence the prediction of their neighbors. While theoretically limiting, such system can still be trained on large graphs without issue, when the receptive field of the network is smaller than the graph diameter (which is a reasonable assumption in knowledge graphs, social networks, etc.). Applications range from citation network labeling to user's preference prediction on social networks or recommender systems of retail websites.

Link prediction: This task is also referred to as graph completion or relational inference. As its name suggests, the task is to infer missing connections in a large, incomplete graph. The setting is similar to node classification, with the difference that the system is trained to predict the presence of edges between pairs of nodes. Applications in social networks can be the recommendation of new connections, pages, content, which are evaluated as appropriate for a given user.

Community detection: Community detection is the task of identifying clusters of nodes that belong to the same category, *i.e.*, community. This task can be trained in a supervised setting with ground truth node-level labels (similar to node classification), or in an unsupervised manner by computing a graph partitioning.

All these tasks require to be able to build node- and graph-level embeddings that encode graph attributes and topological patterns in a *unified* way.

1.3 Graph Neural Networks

1.3.1 The need for deep graph networks

A question that emerges when modeling graph-structured data with neural networks is to understand whether existing architectures can work on graphs, as well as to grasp their limitations.

A first widely-employed class of neural networks are Recurrent Neural Networks (RNNs) that are designed to operate on sequences. A sequence is a special type of graph, called a directed path graph, that can be represented such that all its nodes and (directed) edges lie on a single straight line. This type of graphs implicitly assumes a pre-defined ordering of its nodes. This does not hold in the generic case, where the nodes are not numbered and ordered. Therefore, RNNs can be used to model certain types of graphs, *i.e.*, directed path graph, or when graphs can be *approximated* by directed path graphs, *e.g.*, in chemistry, molecular graphs can be transformed into sequences using the SMILE representation (Weininger et al., 1988), and further processed by a RNNs (Schwaller et al., 2018). RNNs are said to enforce a *sequential* inductive bias in the network.

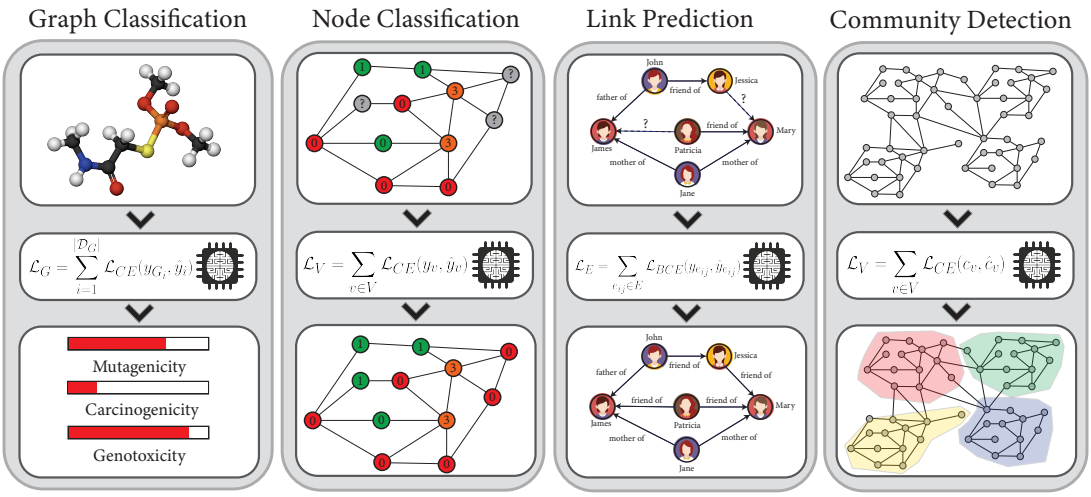


Figure 1.1 – Overview of the main deep graph learning tasks. Graph classification, or regression, learns graph-level representations for predicting graph-level properties. Node classification, or regression, operates in a semi-supervised setting where some unknown node labels need to be inferred from known ones. Link prediction predicts missing connections in an incomplete graph. Community detection identifies clusters of similar nodes according to the graph topology and optional node- and edge-attributes.

Another wide-spread neural network architecture are CNNs. CNNs were initially designed to operate on images, and more generally on grid-like structures. Grids can also be seen as graphs, with a fixed node neighborhood where each node is connected to its eight closest neighbors. Fixed node neighborhoods allow to apply a fixed-size convolutional kernel to the entire grid, therefore inducing a *local* inductive bias in the network. Naturally, such property does not hold in general for graphs. In Section 1.4, we will see how 2D convolution defined on grids can be extended to graphs with *arbitrary* node neighborhoods.

Finally, feed-forward neural networks, or Multi-Layer Perceptrons (MLPs), operate on vectorized inputs and induce a *weak* inductive bias to the network by building all-to-all connections between the input features. MLP input features end up being all "connected", and can be seen as a fully-connected graph.

These considerations highlight that (i) existing architectures are insufficient to learn on arbitrary graph structures, hence motivating the need to develop a novel class of neural networks, and (ii) developing a generic deep neural framework will *generalize* some existing neural architectures, as all the aforementioned data structures can be represented as graphs.

1.3.2 *Desiderata* for a neural graph model

We provide a list of properties that a neural graph model should fulfil to efficiently learn on graph-structured data.

1. **Permutation invariant:** The node ordering of a graph is arbitrary, re-ordering them *does not* change the graph itself, but only its representation. Therefore, a graph model should be invariant to node permutation, thus ensuring that *different* graph representations provide the *same* graph embedding.

2. **Scalable and adaptive:** A graph model should be scalable to an arbitrary large input graph, with an arbitrary number of nodes, edges, node-, and edge-attributes. Moreover, *all* the graphs (as defined in Definition 1.1), *i.e.*, w/ and w/o directed edges, w/ and w/o node- and edge-attributes, should be able to be encoded by the same type of model, *i.e.*, only minor architectural changes should be needed to adapt to different graph types. Also, no prior knowledge beyond the mathematical description of the graph should be required to train the graph model, *i.e.*, the model should remain application-agnostic.

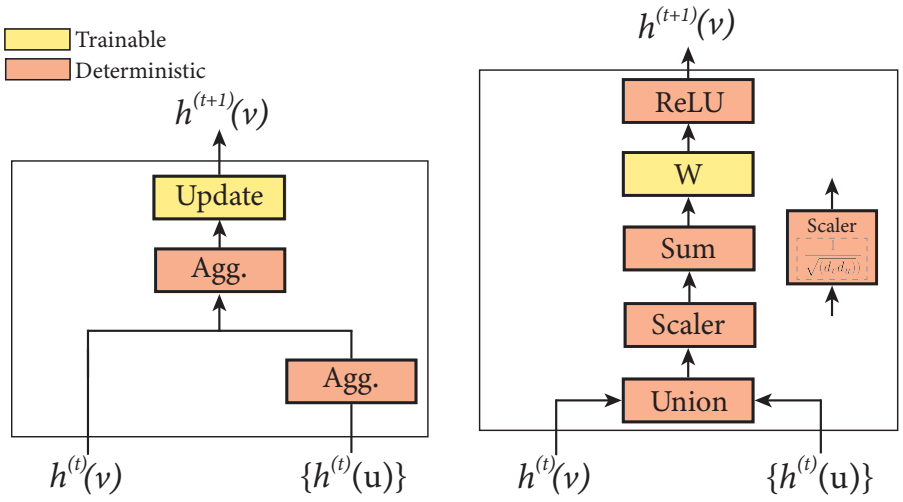
3. **Local:** A graph model should follow a locality principle that states that nearby nodes and edges share more information than distant ones. Intuitively, a graph model should aggregate information from local topological patterns, similarly to the concept of convolution in image representation learning. The connection between CNNs and GNNs will be discussed in this chapter. To build arbitrary deep networks, a graph model should also be composed of layers that can be stacked, thus increasing the network receptive field.

4. **Encode *all* graph properties:** A graph model should leverage all the information encoded in the graph, *i.e.*, the graph adjacency that encodes the graph topological properties along with the graph attributes. Both information should be jointly encoded in a single neural network.

1.3.3 Message Passing Neural Networks

We are first presenting MPNNs (Gilmer et al., 2017), a generic framework to build node embeddings of node-attributed graphs. Then, we discuss its generalization to generic attributed graphs. The theoretical foundations justifying this formulation will be presented in Section 1.4 and in Chapter 2. As we present the framework, we put it in relation with the aforementioned list of graph model *desiderata*.

The node features $h(v)$, $\forall v \in V$ are iteratively updated via a two-step procedure, denoted as the i) AGGREGATE, and ii) UPDATE steps. In the AGGREGATE step for node v, the features of neighboring nodes $\mathcal{N}(v)$ are aggregated into a single feature representation, denoted as $a(v)$. In order to be invariant to node permutation (see *Desideratum* 1), the AGGREGATE step is chosen to be a permutation invariant function, *e.g.*, a sum, mean, etc. In the UPDATE step, the node embeddings of node v are updated by using the current node embeddings and the aggregated features from the AGGREGATE step. Typically, the UPDATE step will be a trainable feed-forward neural network. This step is building *local* (see *Desideratum* 3) representations by jointly encoding the graph topology and attributes, herein fulfilling *Desideratum* 4. A series of T such iterations, denoted as GNN layers, are employed to obtain updated node



(a) MPNN (b) GCN

Figure 1.2 – Overview of MPNN (left) and the GCN (right).

embeddings $\forall\, v \in V$, incorporating information up to T-hops from each node. Therefore, by increasing the number of layers, we increase the receptive field of the network, analogously to CNNs, and addressing *Desideratum* 3. Finally, we build a fixed-size graph-level embedding, denoted as h_G, by pooling the node features $h^{(T)}(v)$ in a READOUT step. Naturally, this step is only employed for graph classification tasks, where a graph embedding is needed. Similarly to the AGGREGATE step, the READOUT needs to be a permutation invariant function. In this way, the algorithm can provide graph embeddings of the same dimension, irrespective of the graph size, in accordance with *Desideratum 2*. To allow for back-propagation and GNN training, the AGGREGATE, UPDATE, and READOUT operations must be differentiable. Formally, the three steps are presented as,

$$a^{(t+1)}(v) = \text{AGGREGATE}\left(\{h^{(t)}(u) : u \in \mathcal{N}(v)\}\right) \tag{1.1}$$

$$h^{(t+1)}(v) = \text{UPDATE}\left(h^{(t)}(v), a^{(t+1)}(v)\right) \tag{1.2}$$

$$h_G = \text{READOUT}\left(\{h^{(T)}(v) : v \in V\}\right) \tag{1.3}$$

Figure 1.2 (left) presents an overview of the MPNN framework by highlighting a node update iteration.

A straightforward MPNN is to use a sum as AGGREGATE and READOUT, and a shallow MLP as UPDATE function, which can be expressed as:

$$h^{(t)}(v) = \sigma\Big(h^{(t-1)}(v) + \sum_{u \in \mathcal{N}(v)} h^{(t-1)}(u)\Big) W^{(t)} \tag{1.4}$$

where σ is the Rectified Linear Unit (ReLU) activation function, $W^{(t)} \in \mathbb{R}^{d^{(t)} \times d^{(t+1)}}$ are train-

able parameters, and $d^{(t)}$, $d^{(t+1)}$ are the node embedding dimensions at layer t and $t+1$, respectively.

1.3.4 Generalized Message Passing

MPNNs are widely used and most of the popular GNN architectures can be expressed with this framework, *e.g.*, Kipf and Welling (2017); Xu et al. (2019b); Morris et al. (2018); Hamilton et al. (2017); Velickovic et al. (2018). These architectures share in common that they operate at node-level. However, graphs can also include edge- and graph-level information that should be modeled by the GNN. To address this limitation, and fulfil *Desiderata 2*, Battaglia et al. (2018) proposed the generalized message passing where the edges and the graph are also represented by embeddings that are updated at each layer. Formally, the generalized MPNN is expressed as:

$$h_E^{(t+1)}(u,v) = \textsc{Update}_E\Big(h_E^{(t)}(u,v), h^{(t)}(u), h^{(t)}(v), h_G^{(t)}\Big) \tag{1.5}$$

$$a^{(t+1)}(v) = \textsc{Aggregate}\Big(\{h_E^{(t+1)}(u,v) : u \in \mathcal{N}(v)\}\Big) \tag{1.6}$$

$$h^{(t+1)}(v) = \textsc{Update}_V\Big(h^{(t)}(v), a^{(t+1)}(v), h_G^{(t)}\Big) \tag{1.7}$$

$$h_G^{(t+1)} = \textsc{Update}_G\Big(h_G^{(t)}, \{h_E^{(t+1)}(u,v),\ \forall (u,v) \in E\}, \{h^{(t+1)}(v),\ \forall v \in V\}\Big) \tag{1.8}$$

An overview of the generalized MPNN is provided in Figure 1.3. Chapter 2 will present a GNN that takes inspiration from this formalism.

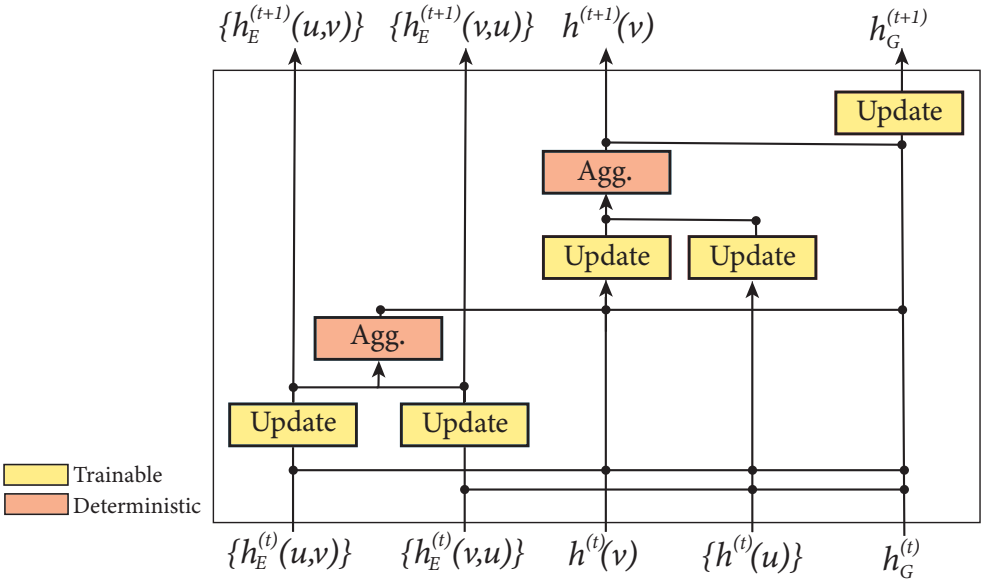


Figure 1.3 – Overview of the generalized MPNN framework.

1.4 Theoretical foundations of GNNs

MPNNs and GNNs can be justified and derived from three different theoretical standpoints. First, GNNs were developed based on graph signal processing (Shuman et al., 2013; Bronstein et al., 2017; Ortega et al., 2018), as a generalization of the convolution operation to graphs (Bruna et al., 2014; Defferrard et al., 2016; Kipf and Welling, 2017). In parallel, Dai et al. (2016) established a parallel between MPNNs and the message passing algorithm used for probabilistic inference in graphical models, *i.e.*, believe propagation. Finally, Hamilton et al. (2017) and Kipf and Welling (2017) highlighted the similarities between GNNs and the Weisfeiler-Lehman test of graph isomorphism. This connection was further studied by Morris et al. (2018); Xu et al. (2019b); Jaume et al. (2019) to characterize the expressivity of GNNs (see Chapter 2). In this section, we provide the key steps that led to the Graph Convolutional Network (GCN) formulation from the graph spectral theory.

The GCN aims to extend the concept of *convolution* to graphs, with the objective to enforce a locality principle with locally spatialized operations. Our starting point is the property that a convolution in the *spatial* domain corresponds to a *multiplication* in the spectral domain. Therefore, we need to define a spectral transformation of graphs.

Formally, let us define the (continuous) convolution operation of two functions f and g as:

$$(f \star g)(\mathbf{x}) = \int_{\mathbb{R}^n} f(\mathbf{x}) g(\mathbf{x}-\mathbf{y}) d\mathbf{y} \tag{1.9}$$

where $\mathbf{x}$ and $\mathbf{y}$ are n-dimensional vectors. One of the main results of signal processing is that convolution can be defined using the *Fourier transform* as:

$$(f \star g)(\mathbf{x}) = \mathscr{F}^{-1}(\mathscr{F}(f(\mathbf{x}))\mathscr{F}(g(\mathbf{x}))) \tag{1.10}$$

where the Fourier transform is defined as $\mathscr{F}(f(\mathbf{x})) = \hat{f}(\xi) = \int_{\mathbb{R}^n} f(\mathbf{x}) \exp(-2\pi \mathbf{x}^T \xi i) d\mathbf{x}$. In signal processing terminology, convoluting a signal f by another h can be seen as filtering the individual elements in $f(\mathbf{x})$ by h.

To apply this principle to graphs, we introduce the laplacian operator defined as $L = D - A$, where D is the diagonal degree matrix computed as $d_{ii} = \sum_j d_{ij}$ and A is the graph adjacency matrix. Intuitively, the multiplication of a signal by the laplacian corresponds to computing the difference between the signal at a node v and its neighbors $\mathscr{N}(v)$. A central result of graph signal processing theory is that the eigenfunctions of the laplacian are the same as the frequency modes of the Fourier transform (Shuman et al., 2013). Therefore, we can generalize the Fourier transform of a graph by looking at the eigendecomposition of the graph laplacian. Formally, by taking an eigenvalue decomposition of L, we obtain the "frequencies" of the graph, which can be expressed as $L = U^T \Lambda U$, where U are the eigenvectors and Λ is a diagonal

matrix with the eigenvalues on its diagonal. The convolution of a function $f \in \mathbb{R}^{|V|}$ by a filter h is then given by:

$$f \star h = U(U^T f U^T h) \tag{1.11}$$

While this formulation is valid, computing the eigenvalue decomposition is expensive and does not scale to large graphs. To overcome this limitation Defferrard et al. (2016) proposed to approximate it with a Chebyshev polynomials expansion as:

$$f \star h \approx \sum_{k=0}^{K} \theta'_k T_k(\tilde{L}) h \tag{1.12}$$

where $\tilde{L} = \frac{2}{\lambda_{max}} L + I_N$, $\theta' \in \mathbb{R}^K$ and T_k is the Chebyshev polynomial of order k. Kipf and Welling (2017) proposed to further simplify this formulation by using 1-hop convolutions, *i.e.*, by setting $K = 1$, and to apply several layers, thereby increasing the receptive field. We end up with the GCN formulation that defines how node embeddings are updated:

$$H^{(t+1)} = \sigma(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}} H^{(t)} W^{(t)}) \tag{1.13}$$

where σ is an activation function (*e.g.*, ReLU), $W^{(t)}$ is the weight matrix at layer t, $\tilde{A} = A + I_{|V|}$, $\tilde{D} = \sum_j \tilde{A}_{ij}$ and $H^{(t)} \in \mathbb{R}^{|V| \times d}$ are the node features at layer t.

Equation (1.13) can be rewritten at the node-level as:

$$h^{(t+1)}(v) = \sigma\Big(\sum_{u \in v \bigcup \mathcal{N}(v)} \frac{1}{d_u d_v} h^{(t)}(u) W^{(t)} \Big) \tag{1.14}$$

A block diagram of the GCN is provided in Figure 1.2. We observe that Equation (1.14) is a particular case of MPNN, where the AGGREGATE step is a degree normalized sum, and the UPDATE step is a one-layer feed-forward neural network.

In this chapter, we provided an overview of deep graph learning, the MPNN framework, and theoretical justifications based on graph spectral theory. The reader should have acquired necessary prerequisites about graphs and GNNs to confidently understand the graph-related contributions of this thesis.

2 Expressitivity of Graph Neural Networks

The ideas, methods and results presented in this chapter are published in:

- "EDGNN: A Simple and Powerful GNN for Directed Labeled Graphs", **Guillaume Jaume***, An-phi Nguyen*, Maria Rodriguez Martinez, Jean-Philippe Thiran, Maria Gabrani, International Conference on Learning Representations (ICLR) workshop on Representation Learning on Graphs and Manifolds, 2019 (Jaume et al., 2019).

GJ (the author of this thesis) is sharing first co-authorship with AN. The ideas, concepts and experiments were designed by GJ and AN. GJ was responsible for implementing the code used to run the experiments, such as the GNN models, the dataloaders, and the experiment manager. AN defined the mathematical description, GJ and AN derived the proofs. JPT, MRM and NG supervised and supported GJ in organizing his research. The manuscript was written by GJ and AN.

2.1 Introduction

MPNNs define a framework for learning on graph-structured data based on a series of node aggregations and updates. The design choices behind the selection of AGGREGATE, UPDATE and READOUT functions will change what the model *can* or *cannot* learn, which we refer to as model *expressivity*. Powerful models are expected to be expressive, as they can represent any input in a distinct embedding location. In the context of graphs, a GNN is said to be expressive if it can distinguish any pair of non-isomorphic graphs. In other words, if the GNN induces an injective mapping between the space of graphs and some embedding space. Xu et al. (2019b) and Morris et al. (2018) independently proved that certain formulations of MPNNs can be *as powerful as* the Weisfeiler–Lehman (WL) test of graph isomorphism (Weisfeiler and Lehman, 1968). This result is similar, in spirit, to the Universal Approximation Theorem for neural networks (Cybenkot, 1989). In practice, this means that there exist MPNNs able to learn *unique* representations for (*almost*) *all* undirected node-labeled graphs.

In this chapter, we present the main results in Morris et al. (2018), by introducing a GNN that can provably be as powerful as the 1-dimensional WL test of graph isomorphism. Then, we extend this result to *directed* graphs with both *node*- and *edge*-labels. In particular, by extending the theoretical framework provided by Morris et al. (2018), we show that there exist MPNNs as powerful as the 1-dimensional WL algorithm for directed labeled graphs. Although previous work proposed GNNs that can operate on directed node- and edge-labeled graphs, *e.g.,* Li et al. (2016); Niepert et al. (2016); Simonovsky et al. (2017); Beck et al. (2018); Schlichtkrull et al. (2018), we present a theoretically-grounded GNN formulation. This class of graphs is encountered in many applications, including scene graph generation (Xu et al., 2017a; Li et al., 2018b; Zellers et al., 2017), link prediction on knowledge graphs (Schlichtkrull et al., 2018) or molecule classification (Xu et al., 2019b; Morris et al., 2018). Specifically, our contributions are:

- We propose EDGNN, a GNN able to operate on graphs with both node- and edge-labels, and directed edges;
- We show that EDGNN can be as powerful as the (extended) 1-dimensional WL test for directed labeled graphs;
- We experimentally show the power of this new formulation on node and graph classification tasks.

2.2 Theoretical framework

2.2.1 Notation and setup

We re-use the notation introduced in Chapter 1, *i.e.*, a graph G is defined as a pair (V_G, E_G), where V_G is the set of nodes and E_G is the set of edges.

In this work, we are interested in graphs with both *node*- and *edge*-labels, and *directed* edges. We therefore assume that, given a graph G, there exist a node-labeling function $l_V : V_G \to \mathscr{X}$ and an edge-labeling function $l_E : E_G \to \mathscr{Z}$ that assign to each node and edge of G a label from *countable* sets $\mathscr{X}$ and $\mathscr{Z}$, *i.e.*, a set whose cardinality $|\mathscr{X}|$ is a subset of $\mathbb{N}$. For the rest of this paper, we will refer to graphs with node- and edge-labels simply as *labeled graphs*.

As we are dealing with directed graphs, the node neighborhood $\mathcal{N}(v) := \{u \in V \mid (v,u) \in E \vee (u,v) \in E\}$ is split between incoming neighbors $\mathcal{N}^I(v) := \{u \in V \mid (u,v) \in E\}$ and outgoing neighbors $\mathcal{N}^O(v) := \{u \in V \mid (v,u) \in E\}$.

Definition 2.2.1. *Two directed and labeled graphs G and H are* isomorphic *if there exists a bijection $f : V_G \to V_H$ such that $(u,v) \in E_G$ if and only if $(f(u), f(v)) \in E_H$ with $l_{V_G}(v) = l_{V_H}(f(v))$ and $l_{E_G}(u,v) = l_{E_H}(f(u), f(v))$.*

Definition 2.2.2. *We define a* multiset *as an ordered pair, denoted as $X = \{\{S, m\}\}$, where S is the*

underlying set of X that is formed from its distinct elements, and $m : S \to \mathbb{N}_{\geq 1}$ *is the multiplicity of each element in S.*

2.2.2 The Weisfeiler–Lehman algorithm

The Weisfeiler–Lehman (WL) test (Weisfeiler and Lehman, 1968) is an algorithm to distinguish whether two graphs are non-isomorphic. We present the test in its *1*-dimensional variant, also known as the *naive vertex refinement*. We will start by presenting the WL test on node-labeled graphs, and later discuss its extension to directed labeled graphs. Finally, we discuss its generalization to k-dimensions, *i.e.*, k-dimensional WL test.

1-dimensional WL test of node-labeled graphs

The goal is to define an algorithm that can discriminate isomorphic from non-isomorphic graphs. The test processes as follows. At initialization, the nodes are labeled consistently with the node-labeling function l_V. We call this the *initial coloring* of the graph and we denote it as $c_l^{(0)}(v) := l_V(v),\ \forall v \in V$. The algorithm then proceeds in a recursive fashion. At iteration t, new labels are computed for each node from the current labels of the node itself and its neighbors, *i.e.*,

$$c_l^{(t+1)}(v) = g\Big(c_l^{(t)}(v), \{\{c_l^{(t)}(u) : u \in \mathcal{N}(v)\}\}\Big), \tag{2.1}$$

where g is an injective hashing function. Each iteration is performed in parallel for the two graphs to be tested, G and H. If at some iteration t, the number of nodes assigned to a label $l \in \mathcal{X}$ differs for the two graphs, then the algorithm stops, concluding that the two graphs are not isomorphic. Otherwise, the algorithm will stop whenever a *stable coloring* is achieved, *i.e.*, whenever $c_l^{(t)}(v_G) = c_l^{(t)}(v_H)$ for all $t \geq T$ and for any pair (v_G, v_H) with $v_G \in V_G$, $v_H \in V_H$, and $c_l^{(T)}(v_G) = c_l^{(T)}(v_H)$. This is guaranteed to happen at most after $T = \max\{|V_G|, |V_H|\}$ iterations. In this case, G and H are considered isomorphic. Figure 2.1 exemplifies the node coloring process used in the WL test.

Even tough the WL test is able to distinguish a wide range of graphs, there exists a class of (fully-characterized) graphs that cannot be discriminated by the WL test. The reader can refer to Cai et al. (1992) for a detailed description. Some efficient implementations were also introduced, for instance Grohe et al. (2017) proposed a flavor with quasi-linear runtime complexity w.r.to the number of nodes.

1-dimensional WL test of labeled graphs

The extension of the WL test to a directed graph with edge labels is straightforward (Grohe et al., 2017; Orsini et al., 2016). During the recursive step, for each node v, we need to include the in-degree and out-degree of v separately in the hashing function w.r.to *each* edge label.

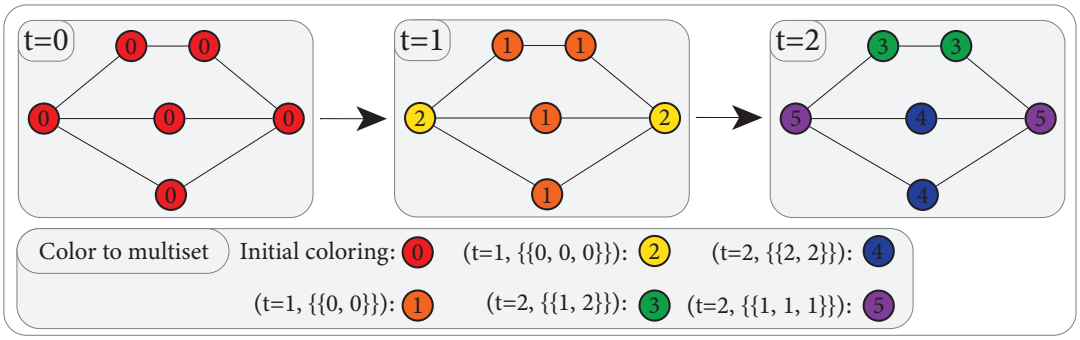


Figure 2.1 – Example of the 1-dimensional WL test in an undirected node-labeled graph. At step $t = 0$, the node colors are assigned to the node labels (all 0s in this example). At step $t = 1$, new colors are assigned according to the neighborhood of each node, *i.e.*, four nodes have two neighbors with label 0, and two with three neighbors with label 0. At step $t = 2$, this procedure is producing a stable coloring, and the algorithm stops.

Let us denote an edge label as $e \in \mathcal{Z}$. For each node v, we then define $n_v^I(e) := |\{u \in \mathcal{N}^I(v) \mid l_E(u,v) = e\}|$ as the number of edges incoming to v with label e. Similarly, $n_v^O(e)$ is defined for outgoing edges. Then, Equation (2.1) can be adapted to directed labeled graphs as:

$$c_l^{(t+1)}(v) = g\Big(c_l^{(t)}(v), \{\{c_l^{(t)}(u) : u \in \mathcal{N}(v)\}\}, \\ \{\big(n_v^I(e), e\big) : \exists (u,v) \in E \text{ with } l_E(u,v) = e\}, \\ \{\big(n_v^O(e), e\big) : \exists (v,u) \in E \text{ with } l_E(v,u) = e\}\Big). \tag{2.2}$$

The rest of the algorithm is executed in the same way as in the standard scenario.

k-dimensional WL test

While this work focuses on the *1*-dimensional WL test, high-order variants exist, referred to as k-dimensional WL tests, that are strictly more powerful than the *1*-dimensional version. It can be shown that the pairs of graphs that can be discriminated by the k-dimensional WL test form a superset of the ones that are discriminated by the $(k-1)$-dimensional version (Kiefer et al., 2020). Specifically, higher-order tests color *k-tuples* of nodes, instead of nodes themselves (see Morris et al. (2018)). The hashing function is modified accordingly to operate at tuple-level. While gaining in expressive power, the algorithm complexity is increasing with k. The reader can refer to Kiefer et al. (2020) for a thorough description.

2.2.3 Provably powerful GNNs

Graph neural networks architectures implement a neighborhood aggregation strategy. Several designs exist, where different AGGREGATE, UPDATE, READOUT functions (satisfying the properties presented in Chapter 1) lead to different model expressivities. In other words, the

GNN architectural properties influence the resulting functions it can represent and learn. In particular, Morris et al. (2018) studied the expressivity of the GNN with a node update function defined as:

$$h^{(t+1)}(v) = \sigma\Big(h^{(t)}(v)W_1^{(t)} + \sum_{u \in \mathcal{N}(v)} h^{(t)}(u)W_2^{(t)}\Big) \qquad (1\text{-GNN})$$

where $h^{(t)}(v) \in \mathbb{R}^{d^{(t)}}$ and $W_1^{(t)}, W_2^{(t)} \in \mathbb{R}^{d^{(t)} \times d^{(t+1)}}$ are weight matrices. Note that the initial representation $h^{(0)}(v)$ is set to be *consistent* with the node-labeling function l_V, *i.e.*, $h^{(0)}(v) = h^{(0)}(u)$ if and only if $l_V(v) = l_V(u)$ for all $v, u \in V$. This GNN flavor, referred to as *1*-GNN, was initially proposed by Hamilton et al. (2017).

GNN expressivity in node-labeled graphs

Theorem 2.2.1 (Theorem 1 in Morris et al. (2018))**.** *Let G be a node-labeled graph. Then for all $t \geq 0$ and for all choices of initial colorings $h^{(0)}$ consistent with l_V, and weights $W_1^{(t)}, W_2^{(t)}$*

$$c_l^{(t)}(v) = c_l^{(t)}(u) \Rightarrow h^{(t)}(v) = h^{(t)}(u) \;\; \forall u, v \in V \qquad (2.3)$$

with $c_l^{(t)}$ and $h^{(t)}$ defined in Equation (2.2) *and Equation* (*1*-GNN)*, respectively.*

In other words, the GNN described by Equation (*1*-GNN) cannot have more expressive power in terms of being able to discriminate between non-isomorphic graphs than the *1*-dimensional WL algorithm.

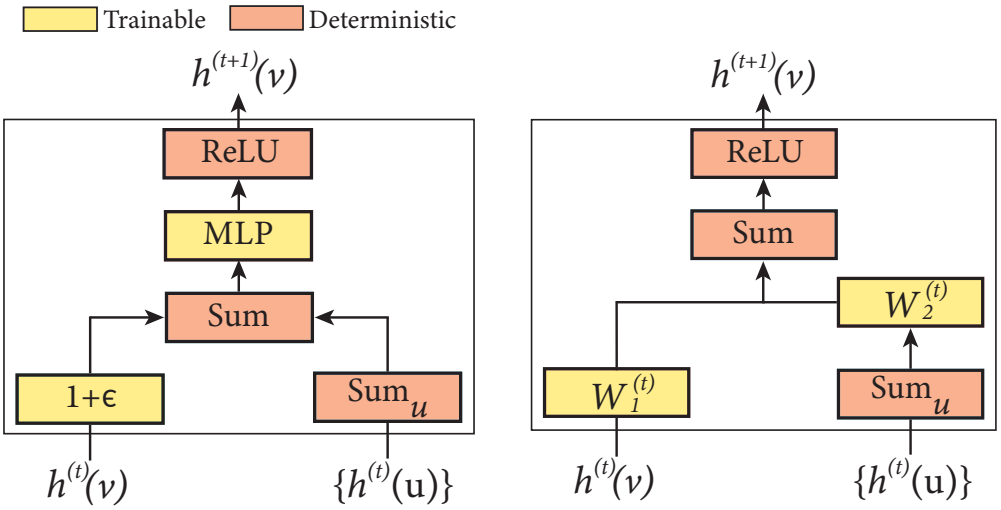
Theorem 2.2.2 (Theorem 2 in Morris et al. (2018))**.** *Let G be a node-labeled graph with finite node degree. Then there exists a sequence $(W_1^{(t)}, W_2^{(t)})$ with $t \geq 0$ such that*

$$c_l^{(t)}(v) = c_l^{(t)}(u) \Leftrightarrow h^{(t)}(v) = h^{(t)}(u) \;\; \forall u, v \in V \qquad (2.4)$$

Which translates to the observation that there exists a sequence of parameters $(W_1^{(t)}, W_2^{(t)})$ such that the GNN implemented as in Equation (*1*-GNN) has exactly the same expressive power as the *1*-dimensional WL test.

The reader can refer to the supplementary material of Morris et al. (2018) for detailed proofs. Intuitively, the proof is based on the relation between a WL test iteration and a GNN layer. The hashing function in Equation (2.1) is "replaced" by the projections induced by the weights $(W_1^{(t)}, W_2^{(t)})$, that can, by using the Universal Approximation Theorem (Cybenkot, 1989) of neural networks, approximate the hashing function.

The results derived by Morris et al. (2018) were also concurrently obtained by Xu et al. (2019b), that proposed the Graph Isomorphism Network (GIN) model, an alternative GNN with the



(a) GIN (b) *1*-GNN

Figure 2.2 – Overview of a GNN and *1*-GNN layer, two architectures that can be as powerful as the *1*-dimensional WL test of graph isomorphism.

same expressive power. Specifically, the GIN node update function is defined as:

$$h^{(t+1)}(v) = \text{MLP}\Big((1+\epsilon^{(t)})h^{(t)}(v) + \sum_{u \in \mathcal{N}(v)} h^{(t)}(u)\Big), \tag{GIN}$$

where $\epsilon^{(t)}$ is an optional trainable parameter.

An overview of the GIN and *1*-GNN architecture is provided in Figure 2.2.

Extension to directed labeled graphs

The extension of the (*1*-GNN) to directed labeled graphs follows the WL test extension. We need to augment the equation with embeddings for the labeled edges with incoming and outgoing edges considered separately, *i.e.*,

$$\begin{aligned} h^{(t+1)}(v) = \sigma\Big(h^{(t)}(v)W_1^{(t)} + \sum_{u \in \mathcal{N}(v)} h^{(t)}(u)W_2^{(t)} + \\ + \sum_{(u,v)\in E} h_E\big(l_E(u,v)\big)W_3^{(t)} + \sum_{(v,u)\in E} h_E\big(l_E(v,u)\big)W_4^{(t)}\Big), \end{aligned} \tag{2.5}$$

where $h_E\big(l_E(v,u)\big) \in \mathbb{R}^{d_E}$ is the d_E-dimensional embedding of the edge (v,u) with label $l_E(v,u)$. The embeddings h_E should be defined such that $\sum_{(u,v)\in E} h_E\big(l_E(u,v)\big) = \sum_{(u,v')\in E} h_E\big(l_E(u,v')\big)$ if and only if $\{\big(n_v^I(e), e\big) : \exists (u,v) \in E \text{ with } l_E(u,v) = e\} = \{\big(n_{v'}^I(e), e\big) : \exists (u,v') \in E \text{ with } l_E(u,v') = e\}$. The same should hold for outgoing edges. In practice, this can be achieved by using a one-hot encoding of the edge labels. An EDGNN layer is presented in Figure 2.3.

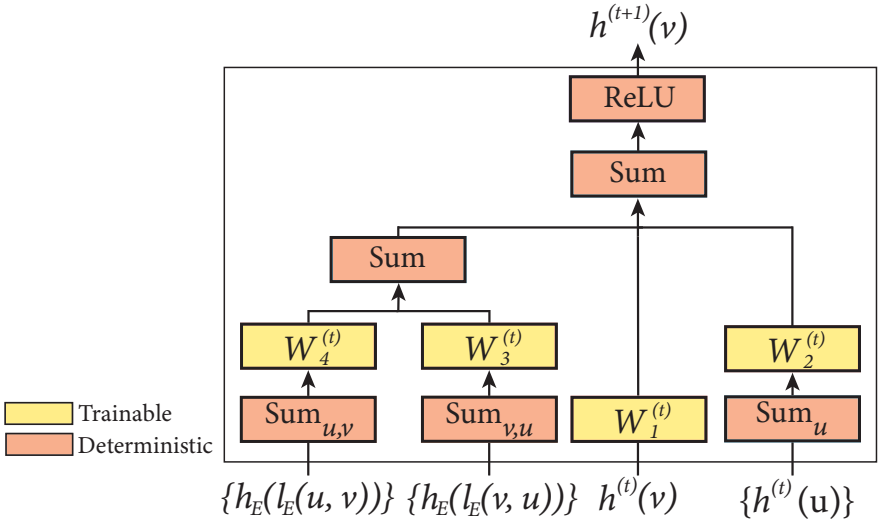


Figure 2.3 – Overview of the EDGNN architecture.

We can now extend Theorem 2.2.1 and Theorem 2.2.2 to directed labeled graphs.

Theorem 2.2.3 (Extension of Theorem 1 in Morris et al. (2018))**.** *Let G be a directed labeled graph. Then for all $t \geq 0$ and for all choices of initial colorings $h^{(0)}$ consistent with l_V and of edge embeddings f_E consistent with l_E, and weights $W_1^{(t)}, W_2^{(t)}, W_3^{(t)}, W_4^{(t)}$*

$$c_l^{(t)}(v) = c_l^{(t)}(u) \Rightarrow h^{(t)}(v) = h^{(t)}(u) \;\; \forall u, v \in V \tag{2.6}$$

with $c_l^{(t)}$ and $h^{(t)}$ defined in Equation (2.2) *and Equation* (2.5)*, respectively.*

Morris et al. (2018) proved this theorem by induction. The proof is essentially the same for our extended case. In fact, as neither the labels l_E nor the embeddings h_E change over the iterations, there is no need to include them in the induction step.

Theorem 2.2.4 (Extension of Theorem 2 in Morris et al. (2018))**.** *Let G be a directed labeled graph with finite node degree. Then there exists a sequence $(W_1^{(t)}, W_2^{(t)}, W_3^{(t)}, W_4^{(t)})$ with $t \geq 0$ such that*

$$c_l^{(t)}(v) = c_l^{(t)}(u) \Leftrightarrow h^{(t)}(v) = h^{(t)}(u) \;\; \forall u, v \in V \tag{2.7}$$

Proof: For a given node v, we define $E_v^I := \{l_E(u,v) : \exists (u,v) \in E\}$ as the set of labels of edges incoming into the node v. We then define $L_v^I := \text{Concat}\Big(\big\{\big(n_v^I(e), e\big) : \forall e \in E_v^I\big\}\Big)$. That is, for each node v, we create a label by concatenating all the labels of the incoming edges together with their multiplicities $n_v^I(e)$. Similarly, we define E_v^O and L_v^O for the outgoing edges. Note that the pairs $\big(n_v^I(e), e\big)$ and $\big(n_v^O(e), e\big)$ take values in $\mathscr{L} := \mathbb{N} \times \mathscr{X}$. Therefore, L_v^I (or L_v^O, respectively) can take values in $\mathscr{L}^{|E_v^I|}$ (or $\mathscr{L}^{|E_v^O|}$, respectively), where $\times$ denotes the Cartesian product.

For all nodes v, we can construct a function $g_v : \mathscr{X} \times \mathscr{L}^{|E_v^I|} \times \mathscr{L}^{|E_v^O|} \to \mathscr{Y}_v$ that *bijectively* maps a tuple $\left(c_l^{(t-1)}(v), L_v^I, L_v^O\right)$ to a label $y \in \mathscr{Y}_v$.

Note that, as we are considering graphs with finite node degree, $|\mathscr{N}^I(v)|$ and $|\mathscr{N}^O(v)|$ (and, consequently, $|E_v^I|$ and $|E_v^O|$) are finite. Therefore, $\mathscr{X} \times \mathscr{L}^{|E_v^I|} \times \mathscr{L}^{|E_v^O|}$ is a *countable* set because the finite Cartesian product of countable sets is itself countable. Thus, as we built the function g_v to be bijective, the sets $\mathscr{Y}_v$, and their countable union $\mathscr{Y} := \bigcup_{v \in V} \mathscr{Y}_v$, are also countable (for results on countable sets, refer to Patterson et al. (1967)).

We can then construct an injective hash function g' such that

$$g'\left(y, \{c_l^{(t-1)}(u) : u \in \mathscr{N}(v)\}\right) = g\left(c_l^{(t-1)}(v), \{c_l^{(t-1)}(u) : u \in \mathscr{N}(v)\}, L_v^I, L_v^O\right). \tag{2.8}$$

where the right-hand side is the relabeling function defined in Equation (2.2). These constructions highlight the fact that an iteration of the WL algorithm on a directed labeled graph is the same as performing an iteration of the WL algorithm on a undirected node-labeled graph, where node labels take values in an appropriately augmented label set $\mathscr{Y}$.

The same equivalence can be highlighted between the GNN update functions in Equation (*1*-GNN) and Equation (2.5). In fact, Equation (2.5) can be rewritten as

$$h^{(t)}(v) = \sigma\left(h_{\mathscr{Y}}^{(t-1)}(v) W_{1,3,4}^{(t)} + \sum_{u \in \mathscr{N}(v)} h^{(t-1)}(u) W_2^{(t)}\right), \tag{2.9}$$

where $h_{\mathscr{Y}}^{(t)}(v) \in \mathbb{R}^{1 \times (d^{(t)} + 2d_E)}$ is the embedding resulting from the (horizontal) concatenation of $h^{(t)}(v)$, $\sum_{(u,v) \in E} h_E\big(u, v, l_E(u,v)\big)$, and $\sum_{(v,u) \in E} h_E\big(v, u, l_E(v,u)\big)$, whereas $W_{1,3,4}^{(t)}$ is the (vertical) concatenation of $W_1^{(t)}$, $W_3^{(t)}$, and $W_4^{(t)}$.

The reformulations presented in Equation (2.8) and Equation (2.9) allow us to treat our problem as one of undirected graphs with labels only for nodes. We can therefore prove this theorem by directly using the proof of Theorem 2 in Morris et al. (2018). □

Graph classification

For graph classification tasks, a graph-level representation h_G is needed. We build it from the node representations $h^{(t)}(v)$ following the formulation in Xu et al. (2019b):

$$h_G = \text{Concat}\left(\left\{ \sum_{v \in V_G} h^{(t)}(v) \;\middle|\; t = 0, \ldots, T \right\}\right). \tag{2.10}$$

It can be shown (see Xu et al. (2019b)) that Equation (2.10) builds graph embeddings that preserve the expressive power of the GNN. Intuitively, this is guaranteed by the fact that the *sum* operator is injective over the multiset induced by the node embeddings. Note that, although T should theoretically be at least $|V_G|$ (Section 2.2.2), only a few layers (*i.e.*, iterations)

are used in practice to update the node representations. An MLP classifier is finally applied to the graph representation to perform classification (or regression).

2.2.4 Extension to *continuous* node features

In many real-world applications, the node features take values in $\mathbb{R}^n$, and are therefore not drawn from a *countable* set. For instance, CNN-based deep features and location-based features represent *continuous* feature spaces where Theorem 2.2.3 and Theorem 2.2.4 do not hold.

Extensions of the work by Morris et al. (2018) and Xu et al. (2019b) showed that for *continuous* node features, the use of multiple permutation-invariant aggregators, such as *sum* and *max*, allows to increase the model expressiveness (Dehmamy et al., 2019; Corso et al., 2020). To this end, Corso et al. (2020) proposed the Principal Neighborhood Aggregation (PNA) network that employs a combination of *aggregators*, which are generalizing the *sum* operator used in GIN, and *degree-scalers*, used to scale neighboring aggregated-messages according to the node degree. They showed that to discriminate multisets of size n whose underlying set is $\mathbb{R}$, at least n aggregators are needed, where an aggregator is defined as $f : \{\{.\}\} \rightarrow \mathbb{R}$.

Specifically, Corso et al. (2020) propose a node update defined as,

$$\begin{aligned} a^{(t+1)}(v) &= \bigoplus_{u \in \mathcal{N}(v)} M^{(t)}\left(h^{(t)}(v),\, h^{(t)}(u)\right) \\ h^{(t+1)}(v) &= U^{(t)}\left(h^{(t)}(v),\, a^{(t+1)}(v)\right) \end{aligned} \tag{2.11}$$

where $t = 0, \ldots, T$ is the iteration index. As shown in Figure 2.4, for a node v, first, the neighboring node embeddings $\{h^{(t)}(u)\}, \forall u \in \mathcal{N}(v)$ are concatenated with $h^{(t)}(v)$, and processed by $M^{(t)}$, a MLP, to produce a set of neighborhood-aware embeddings. Then, multiple aggregators with degree-scalers denoted by $\bigoplus$ operate on the set of MLP embeddings to extract a set of multivariate information that expresses the neighborhood distribution of node v. Finally, these representations are concatenated to produce the aggregated message $a^{(t+1)}(v)$. Afterwards, $a^{(t+1)}(v)$ and $h^{(t)}(v)$ are concatenated and processed by $U^{(t)}$, a MLP, to update the node embedding $h^{(t+1)}(v)$. Details of $\bigoplus$ are presented as,

$$\begin{aligned} \bigoplus &= \Big[I, \mathcal{S}(D, \alpha = 1), \mathcal{S}(D, \alpha = -1)\Big] \otimes \Big[\mu, \sigma, \max, \min\Big] \\ \mathcal{S}(D, \alpha) &= \frac{\log(D+1)^{\alpha}}{\delta}, \quad \delta = \frac{1}{|V_{train}|} \sum_{v \in V_{train}} \log(d_v + 1) \end{aligned} \tag{2.12}$$

where I is the identity matrix, $\mathcal{S}$ is the degree-scaler matrix, D is the node degree matrix, δ is a normalization constant, α is a scaling variable, and V_{train} represents the set of nodes used for training. $[I, \mathcal{S}(D, \alpha = 1), \mathcal{S}(D, \alpha = -1)]$ and $\Big[\mu, \sigma, \max, \min\Big]$ denote the list of scalers and the list of aggregators, respectively. The aggregators compute statistics about the node neighborhood, and the injective scalers discriminate between the multisets. $\alpha = \{-1, 0, 1\}$

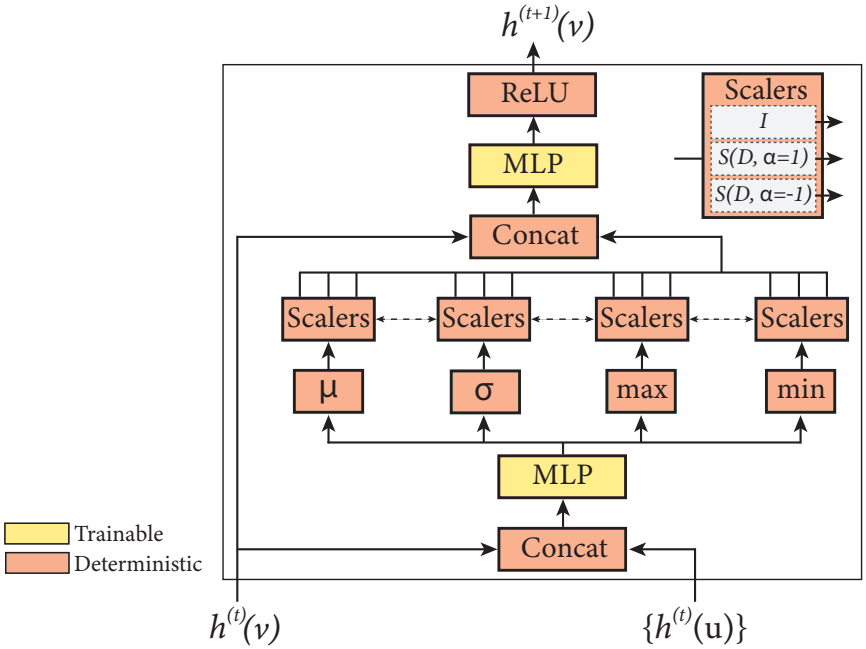


Figure 2.4 – Overview of the PNA architecture.

allows to attenuate, remove, or amplify the scaling. $\otimes$ denotes the tensor product between scalers and aggregators, and produces twelve operations that extract the set of multivariate information.

2.3 Experiments

All the experiments described in this section are conducted on node- and edge-labeled graphs with countable labels to show the power of our proposed method, EDGNN. GNNs based on GIN and PNA will be used in Chapter 5,6,7.

2.3.1 Datasets and baselines

We benchmark our algorithm EDGNN on graph and node classification tasks.

Graph classification

Graph classification is evaluated on two datasets, (i) MUTAG (Debnath et al., 1991; Kriege et al., 2012), a dataset of nitroaromatic compounds, where the task is to predict their mutagenicity on salmonella typhimurium, and (ii) PTC (Helma et al., 2003), a dataset of chemical compounds and their associated carcinogenicity on rats Dataset statistics are shown in Table 2.1.

Dataset	Graphs	Classes	Avg nodes	Avg edges	Node labels	Edge labels
MUTAG	188	2	17.9	19.8	6	3
PTC FM	349	2	14.1	14.5	18	4
PTC FR	351	2	14.6	15.0	19	4
PTC MM	336	2	14.0	14.3	20	4
PTC MR	344	2	14.3	14.7	18	4

Table 2.1 – Graph classification dataset statistics providing the number of graphs (Graphs), the number of classes (Classes), the average number of nodes per graph (Avg nodes), the average number of edges per graph (Avg edges), the number of node labels (Node labels) and the number of edge labels (Edge labels).

We benchmark our proposed model, EDGNN, against the Subgraph Matching Kernel (CSM) (Kriege et al. (2012)), the Weisfeiler–Lehman Shortest Path Kernel (Shervashidze et al. (2011)) and R-GCN (Schlichtkrull et al. (2018)). As R-GCN was designed for node classification tasks, the original paper does not specify how to build graph embeddings. We therefore re-use the formulation in Xu et al. (2019b) to build a graph-level representation.

Graph classification experiments were run with a batch size of 8 and a learning rate of 10^{-4} with 5×10^{-4} weight decay. We then performed a parameter search over the number of layers and node embedding size. The best performance was reached by using two GNN layers with 64 hidden units and ReLU activation. The system was trained for at most 40 epochs with early stopping w.r.to the validation set cross-entropy loss. The GNN was initialized with a one-hot encoding of the node and edge features.

R-GCN (adapted for graph classification) was also trained with a batch size of 8, 10^{-4} learning rate with 5×10^{-4} weight decay. We used three layers with 64 hidden units with learnable nodes embeddings. We used a basis decomposition with the number of basis set to the number of edge types. Results for CSM (Kriege et al., 2012) and WLSP (Shervashidze et al., 2011) are based on the re-implementation of Kriege et al. (2012). All our experiments were performed with 10-fold cross validation as in Kriege et al. (2012).

Node classification

Node classification is tested on two datasets. First, AIFB Ristoski et al. (2016), a semantic Web dataset that represents the organizational structure of the AIFB research institute at the University of Karlsruhe. The task is to predict the research group associated to each person in the institute. And, Mutagenicity, a dataset that contains complex molecules that are potentially carcinogenic, characterized by their mutagenicity.

We benchmark EDGNN against R-GCN (Schlichtkrull et al., 2018), RDF2Vec (Ristoski et al., 2016) and WL (Shervashidze et al., 2011; De Vries et al., 2015)).

Similar to the graph classification setting, we initialize the node and edge features with a one-hot encoding of their input label. When no node label is provided, we use the in-degree. Dataset statistics are presented in Table 2.2.

The node classification experiments were run with a learning rate of 5×10^{-3} without weight decay. We used dropout 0.5 on each layer with ReLU activation. The best performance was achieved by using two GNN layers with 64 hidden units. The maximum number of epochs was set to 400 with early stopping w.r.to the validation set cross-entropy loss. Results with R-GCN (Schlichtkrull et al., 2018), RDF2Vec (Ristoski et al., 2016) and WL (Shervashidze et al., 2011) are based on the re-implementation of Schlichtkrull et al. (2018).

Dataset	Classes	Nodes	Edges	Edge labels
AIFB	4	8,285	29,043	45
MUTAGENICITY	2	23,644	74,227	23

Table 2.2 – Node classification dataset statistics highlighting the number of classes (Classes), the total number of nodes (Nodes), the total number of edges (Edges) and the number of edge labels (Edge labels).

2.3.2 Results and discussion

Table 2.3 presents graph classification average accuracy results together with standard deviation obtained over ten training runs. Our provably powerful model, EDGNN, reaches comparable performance with the state-of-the-art. We observe that the kernel-based and GNN-based methods (R-GCN and EDGNN) perform similarly without being able to clearly identify better models. We conjecture that the relatively small size of the datasets (*e.g.*, only 188 graphs in the MUTAG dataset) does not allow to fully explore the potential of the most expressive models.

Model	MUTAG	PTC FM	PTC FR	PTC MM	PTC MR
CSM	85.4 ± 1.2	$\mathbf{63.8 \pm 1.0}$	65.5 ± 1.4	63.3 ± 1.7	58.1 ± 1.6
WLSP	85.4 ± 1.2	60.4 ± 1.32	65.7 ± 1.3	$\mathbf{66.6 \pm 1.1}$	$\mathbf{59.7 \pm 1.6}$
R-GCN	81.5 ± 2.1	60.7 ± 1.7	$\mathbf{65.8 \pm 0.6}$	64.7 ± 1.7	58.2 ± 1.7
EDGNN (avg)	$\mathbf{86.9 \pm 1.0}$	59.8 ± 1.5	65.7 ± 1.3	64.4 ± 0.8	56.3 ± 1.9
EDGNN (max)	88.8	62.2	68.0	66.1	59.4

Table 2.3 – Graph classification results in accuracy obtained with 10-fold cross validation. Results are expressed as %. EDGNN is compared with the Subgraph Matching Kernel (CSM) (Kriege et al. (2012)), Weisfeiler–Lehman Shortest Path Kernel (Shervashidze et al. (2011)) and R-GCN (Schlichtkrull et al. (2018)).

For node classification (see Table 2.4), EDGNN also achieves comparable performance with the state-of-the-art without outperforming it. This does not contradict our theoretical findings.

In fact, the power of a learnable model does not guarantee its generalization nor that the best model can be learned. However, it is true that, *in the best-case scenario*, a more powerful model should perform better than a less powerful one, as shown by the results regarding the best-learned EDGNN model (max).

Model	AIFB	MUTAG
WL	80.5 ± 0.0	$\mathbf{80.9 \pm 0.0}$
RDF2Vec	88.9 ± 0.0	67.2 ± 1.2
R-GCN	$\mathbf{95.8 \pm 0.6}$	73.2 ± 0.5
EDGNN (avg)	91.1 ± 2.4	80.0 ± 3.2
EDGNN (max)	**97.2**	**85.3**
EDGNN (emb)	91.1 ± 1.7	77.2 ± 2.6
EDGNN (reg)	89.4 ± 1.7	80.4 ± 3.4

Table 2.4 – Node classification results in accuracy averaged over ten runs. Results are expressed as %. EDGNN is compared with WL (De Vries et al. (2015)), RDF2Vec (Ristoski et al. (2016)) and R-GCN (Schlichtkrull et al. (2018)).

2.4 Conclusion

In this chapter, we studied the expressive power of GNNs, by establishing a parallel between message passing and the WL test of graph isomorphism. This study provides the theoretical tools to build various GNN architectures depending on the application at hand. When dealing with node-labeled graphs with *discrete* features, the GIN architecture is a natural design choice. For better expressivity when graphs have *continuous* node labels, PNA can be used. Finally, when graphs have node- and edge-labels, our proposed EDGNN can be as powerful as the *1*-dimensional WL algorithm for graph isomorphism, and has empirically shown to produce promising performance.

3 Interpretability of Graph Neural Networks

The ideas and methods presented in this chapter are partially derived and adapted from:

- "Quantifying Explainers of Graph Neural Networks in Computational Pathology", **Guillaume Jaume***, Pushpak Pati*, Behzad Bozorgtabar, Antonio Foncubierta, Anna Maria Anniciello, Florinda Feroce, Tilman Rau, Jean-Philippe Thiran, Maria Gabrani, Orcun Goksel. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021 (Jaume et al., 2021b).
- "Towards Explainable Graph Representations in Digital Pathology", **Guillaume Jaume***, Pushpak Pati*, Antonio Foncubierta, Florinda Feroce, Giosue Scognamiglio, Anna Maria Anniciello, Jean-Philippe Thiran, Orcun Goksel, Maria Gabrani. In *International Conference on Machine Learning (ICML), ICML Workshop on Computational Biology*, 2020 (Jaume et al., 2020).

GJ (the author of this thesis) is sharing first co-authorship with PP on both publications. A detailed description of each author's contribution is provided in Chapter 6.

3.1 Introduction

Deep learning on graphs has emerged as one of the most active topics in DL. In particular, GNNs, introduced in Chapter 1, have shown to be ideal neural candidates for efficiently learning on graph-structured data. While the development of GNNs has primarily focused on improving performance, *interpretability* of GNNs remains an open research question. Similarly to other popular neural architectures, *e.g.*, CNN, RNN, etc., GNNs are "black-box" models, where the exact process leading to a prediction is too complex to be grasped by humans. This lack of transparency can hinder the deployment of GNNs in real-life settings, especially for applications that demand explainable and trustable predictions. For instance, in a medical setting, trust between a doctor and an AI can only be established if there is a way to interrogate the model to justify its prediction. The model explanation for a given sample

can then take various forms, *e.g.*, a saliency map (Pope et al., 2019; Baldassarre and Azizpour, 2019), or can be expressed as a representative input subset (Ying et al., 2019).

XAI is an extensively studied field in NLP (Danilevsky et al., 2020) and CV (Selvaraju et al., 2017; Chattopadhay et al., 2018). In particular, post-hoc methods, designed to explain intrinsically uninterpretable models, have gained a lot of attention with the development of gradient- (Baldassarre and Azizpour, 2019; Selvaraju et al., 2017), feature- (Zhou et al., 2016), surrogate- (Ribeiro et al., 2016), and decomposition-based (Bach et al., 2015) methods, among others. These methods provide local *instance-level* explanations, *i.e.*, one explanation per sample, that are highlighting the most important parts of the input, *e.g.*, a set of pixels in an image, for making a prediction. This class of algorithms is referred to as *feature attribution* methods. However, when extended to graph-structured data, XAI faces new challenges. In particular, graphs represent complex and entangled relationships between entities, therefore, *evaluating* the explanation relevance is not straightforward. For instance, while a cat-vs-dog classifier focusing on the cat's whiskers will easily convince us as a good explanation, extending it to brain connectivity networks or protein interaction networks requires domain-specific expertise. Additionally, finding the appropriate *units of explanation*, *i.e.*, what is used to define the explanation, of a graph is challenging. In text or image analysis, the *units of explanation* can trivially be defined at word- and pixel-level, respectively. In graphs, deciding if the units should be at node-, edge-, node feature-, or edge feature-level is a design choice, that the explanation model should be able to adapt to.

In this chapter, we propose four graph explainers that provide sample-level explanations, expressed as a set of the most important *nodes* for making the prediction at hand. By focusing on node importance, we reduce the explanation complexity by treating the node features characterizing a node as a single *unit of explanation*. While the proposed methods do not explicitly encode edge-level importance scores, they remain implicitly used as part of the GNN computational graph. Specifically, our contributions are:

- We introduce four graph explainers, GNNEXPLAINER, GRAPHGRAD-CAM, GRAPHGRAD-CAM++, and GRAPHLRP that operate in a *similar* setting and provide *comparable* explanations, thereby allowing for qualitative and quantitative benchmarks;

- We propose GRAPHGRAD-CAM++, an extension of GRAD-CAM++ that can operate on graph-structured data;

- We reformulate the GNNEXPLAINER to be directly applicable to graph classification tasks, instead of node classification. Our formulation allows to build compact explanations, without the need for post-processing, or assuming priors about the task at hand.

3.2 Background

Interpretability is a broad topic in machine learning, with sometimes inconsistent definitions and goals. In this section, we first present a set of *desiderata* related to deep graph interpretability, before providing an overview of existing approaches.

3.2.1 Graph explanation requirements

The goal of deep graph interpretability is to identify the nodes, edges, node features and edge features that are important for making a certain prediction. Informally, an explanation is said to be “good" if the identified subgraph matches our expectations and our own understanding of the task. Specifically, we define four requirements for building graph explainers:

- *Fidelity*: the explanation, *i.e.*, the graph subset, needs to have consistent prediction with the original graph. In other words, processing the original graph or the explanation to the model should lead to the *same* prediction;
- *Sparsity*: the explanation needs to be as small as possible, *i.e.*, we should prune as many graph components while ensuring *fidelity*;
- *Stability*: the graph explainer should provide *similar* explanations for *similar* input graphs, *i.e.*, small modifications to the input graph should only marginally affect the explanation;
- *Accuracy*: the explanation needs to be aligned with the ground truth. However, in many real-world cases, ground truth explanation is not accessible, nor uniquely defined, *e.g.*, several convincing explanations can attest of the presence of a cat in an image. Therefore, we relax this requirement by stating that the explanation needs to match our understanding of the task. The development of robust metrics when ground truth is not available will be further discussed in Chapter 6.

3.2.2 Taxonomy of deep graph learning interpretability

Deep graph interpretability can be defined at instance- or model-level. In instance-level interpretability, a graph explainer identifies important input features of a given query graph, *e.g.*, a node subset, responsible for the prediction. Differently, model-level interpretability aims to extract representative graph patterns that model certain behaviors. In this work, we focus on instance-level methods, that we can further categorize in four groups:

- **Gradient-based methods:** These approaches define node importance by measuring the gradient of an output class, *e.g.*, the predicted class, w.r.to the input or some deep representation. A positive and high gradient value indicates that the query feature has a positive impact on the prediction, while a negative or low gradient value indicates a

negative or no influence, respectively, of that feature (Baldassarre and Azizpour, 2019; Pope et al., 2019). GRAPHGRAD-CAM and GRAPHGRAD-CAM++, presented hereafter, are both gradient-based methods;

- **Perturbation-based methods:** This class of approaches study the effect of small input perturbations on the output. Intuitively, when removing discriminative graph components, the predictions should change, whereas nodes and edges conveying no information should not impact the prediction. By characterizing these changes, one can derive sample-level explanations as proposed in Ying et al. (2019); Luo et al. (2020); Yuan et al. (2020); Schlichtkrull et al. (2021). GNNEXPLAINER and our proposed extension are both perturbation-based methods;

- **Decomposition-based methods:** By decomposing the original model prediction from the predicted logits back to the input features, one can understand the relationship between the input- and prediction-space, and derive feature-level importance scores (Baldassarre and Azizpour, 2019; Pope et al., 2019; Schwarzenberg et al., 2019). GRAPHLRP, which decomposes the output with layerwise relevance propagation rules, belongs to this category;

- **Surrogate methods:** Differently, these approaches explain a (complex) model prediction with a simple and interpretable surrogate model, *e.g.*, a linear model, to approximate the original model prediction around some query sample (Huang et al., 2021; Vu et al., 2020).

The reader can refer to Yuan et al. (2021) for a thorough and detailed review of deep graph interpretability.

3.3 Methods

In this section, we formally present four post-hoc graph explanation techniques: GRAPHLRP, GRAPHGRAD-CAM, GRAPHGRAD-CAM++, and GNNEXPLAINER. These methods form the theoretical foundations of Chapter 6, where (i) we will show the potential of deep graph interpretability for explaining predictions made on histology images, and (ii) we will emphasize on the importance of developing metrics in the absence of ground truth explanations.

3.3.1 Notation

Following the notation introduced in Chapter 1, we define an attributed graph $G := (V, E, H)$ as a set of nodes V, edges E, and node attributes $H \in \mathbb{R}^{|V| \times d}$. d denotes the number of attributes per node, and $|.|$ denotes set cardinality. The graph topology is defined by a graph adjacency matrix, $A \in \mathbb{R}^{|V| \times |V|}$, where $A_{uv} = 1$ if $(u, v) \in E$. $H_{n,k}$ expresses the k-th attribute of the n-th node. The forward prediction of a graph G is denoted as, $y = \mathscr{M}(G)$, where $\mathscr{M}$ is a model operating on graphs, and $y \in \mathbb{R}^{|\mathscr{C}|}$ are output logits. Notation $y(c)$, $c \in \mathscr{C}$ denotes the output

logit of the c-th class. We refer to the logit of the predicted class as $y_{\max} = \max_{c \in \mathscr{C}} y(c)$, and the predicted class as $k_{\max} = \operatorname{argmax}_{c \in \mathscr{C}} y(c)$.

3.3.2 Graph explainer setting

Each graph explainer operates in a similar setting. Namely,

- The input is a node-attributed graph G, as defined in Chapter 1;
- We assume that a model $\mathscr{M}$ was trained *a priori* and can be used for inference. In the sequel, we assume $\mathscr{M}$ is a GNN model as introduced in Chapter 1. Note that different graph learning models could also be combined with the presented graph explainers, but this is beyond the scope of this work;
- An explanation is always generated by explaining the contribution of a single logit, denoted as the query logit *e.g.*, the predicted class $y_{\max}$;
- Each explainer returns normalized node-level importance scores that characterizes the relevance of each node for classifying a certain class, *e.g.*, for classifying $t_{\max}$;
- Node importance scores can be thresholded to define the graph *explanation*, denoted as $G_s = (V_s, E_s, H_s) \subset G$. The explanation graph topology is trivially derived by keeping all the edges connected to the remaining nodes, *i.e.*, $E_s = \{(u,v) | u, v \in V_s\ , (u,v) \in E\}$.

3.3.3 Layerwise relevance propagation: GRAPHLRP

Layerwise Relevance Propagation (LRP) (Bach et al., 2015) is a decomposition-based method. LRP explains an output logit, defined as the *relevance* of a class, by *decomposing* the individual contributions of each input element. LRP was initially formulated for operating on fully connected layers (LRP-FC), and works as follows. Given a pre-trained weight matrix $W \in \mathbb{R}^{z_1 \times z_2}$ between layer 1 and layer 2, where z_1 and z_2 are the number of neurons in layer 1 and layer 2, respectively, we define the z^+ propagation rule (Montavon et al., 2015) that back-propagates the *positive* neuron contributions from layer 2 to layer 1 as:

$$R_i = \sum_j^{z_2} \frac{f_i |w_{ij}|}{\sum_k^{z_1} f_k |w_{kj}|} R_j \tag{LRP-FC}$$

where $|w_{ij}|$ is the absolute value of the weight between i-th and j-th neuron in layer 1 and 2, respectively, and f_i denotes the activation of the i-th neuron in layer 2.

The extension from LRP-FC to LRP for GNNs (GRAPHLRP) is achieved by following the observations in Schwarzenberg et al. (2019). The *aggregate step* in a GNN corresponds to projecting the graph's adjacency matrix on the node embedding space. Assuming a GNN of the form:

$$H^{(t+1)} = \sigma\left(W^{(t)}(I + \tilde{A})H^{(t)}\right) \tag{3.1}$$

where $\tilde{A}$ is the degree-normalized graph adjacency matrix, *i.e.*, $\tilde{A}_{ij} = \frac{1}{|\mathcal{N}(i)|} A_{ij}$, σ is the ReLU activation function. The GNN in Equation (3.1) corresponds to a GIN layer with a 1-layer MLP as an update function, and a *mean* aggregator.

This representation allows us to treat the term $(I + \tilde{A})$ as a regular, fully connected layer. We can then apply the z^+ propagation rule with weights w_{ij} defined as:

$$w_{ij} = 1 \quad \text{if } i = j \tag{3.2}$$

$$w_{ij} = \frac{1}{|\mathcal{N}(i)|} \quad \text{if } (i, j) \in E \tag{3.3}$$

$$w_{ij} = 0 \quad \text{otherwise} \tag{3.4}$$

LRP outputs an importance score for each node i in the graph. The final explanation is derived by thresholding node-level importance scores, thereby enforcing explanation *sparsity*.

3.3.4 Gradient-based: GRAPHGRAD-CAM

GRAD-CAM (Selvaraju et al., 2017) is a gradient-based method that identifies salient regions in the input space. It assigns importance scores to each element of the input to produce a Class Activation Map (CAM) (Zhou et al., 2016). While originally developed for explaining CNNs operating on images, GRAD-CAM can be extended to GNNs (Pope et al., 2019).

GRAPHGRAD-CAM processes in two steps. First, it assigns an importance score to each channel of the GNN, *i.e.*, along each node embedding dimension. The importance of channel k in layer t is computed by measuring the gradient intensity of the logit $y(c)$ w.r.to node attributes $H^{(t)}_{n,k}$. Formally expressed as:

$$w^{(t)}_k = \frac{1}{|V|} \sum_{n=1}^{|V|} \frac{\partial y(c)}{\partial H^{(t)}_{n,k}} \tag{3.5}$$

Intuitively, large positive gradient values are evidences of the presence of the class under consideration, while small gradient values have no influence on its presence. A formal mathematical description is presented in Appendix A, where we show that this formulation can be seen as a generalization of CAM (Zhou et al., 2016).

Then, node-wise importance scores are computed using the forward node feature activations $H^{(t)}$ as:

$$L(t, v) = \text{ReLU}\Big(\sum_{k}^{d^{(t)}} w^{(t)}_k H^{(t)}_{n,k} \Big) \tag{3.6}$$

where $L(t, v)$ denotes the importance of node $v \in V$ in layer t, and $d^{(t)}$ denotes the number of node attributes w.r.to layer t. Since we are only interested in the positive node contributions, *i.e.*, nodes that positively influence the class prediction, we apply a ReLU activation to the

node importance scores. Following prior work by Pope et al. (2019), we take the average scores obtained over all the GNN layers to obtain smoother representations, *i.e.*,

$$L(v) = \frac{1}{T} \sum_{t \in \{1,\dots,T\}} L(t,v), \ \forall v \in V \qquad \text{(GRAPHGRAD-CAM)}$$

As in GRAPHLRP, the node-level importance scores can be thresholded to define the explanation.

3.3.5 Gradient-based: GRAPHGRAD-CAM++

GRAPHGRAD-CAM++ extends GRAD-CAM++ (Chattopadhay et al., 2018) to graph-structured data. It improves the node importance localization of GRAD-CAM by introducing node-wise contributions to the channel importance score computation. It builds on the work by Zhou et al. (2016), that empirically proved to have localization properties. Specifically, Equation (3.5) is modified as:

$$w_k^{(t)} = \sum_{n=1}^{|V|} \alpha_{n,k}^{(t)} \text{ReLU}\Big(\frac{\partial y(c)}{\partial H_{n,k}^{(t)}}\Big) \tag{3.7}$$

where $\alpha_{n,k}^{(t)}$ are node-wise weights expressed for each channel k of layer t.

We show that $\alpha_{n,k}^{(t)}$ can be computed as:

$$\alpha_{n,k}^{(t)} = \frac{\frac{\partial^2 y_{\max}}{(\partial H_{n,k}^{(t)})^2}}{2\frac{\partial^2 y_{\max}}{(\partial H_{n,k}^{(t)})^2} + \sum_{n=1}^{|V|} H_{n,k}^{(t)}\big(\frac{\partial^3 y_{\max}}{(\partial H_{n,k}^{(t)})^3}\big)} \tag{3.8}$$

The proof is provided in Appendix A, and is analogous to the derivation proposed in Chattopadhay et al. (2018), where the number of nodes represents the “spatial" dimensions.

The subsequent node importance computation in GRAPHGRAD-CAM++ follows the one in GRAPHGRAD-CAM, *i.e.*, we use Equation (3.6) to derive $L(t,v)$ and Equation GRAPHGRAD-CAM to get the final $L(v)$.

Note that the explanation *fidelity* is implicitly ensured in GRAPHLRP, GRAPHGRAD-CAM and GRAPHGRAD-CAM++ as the retained input elements are the ones used by the network to increase a given logit value.

3.3.6 Graph pruning: GNNEXPLAINER

GNNEXPLAINER is a post-hoc perturbation technique based on graph pruning and originally proposed by Ying et al. (2019). GNNEXPLAINER is model-agnostic and can explain any flavor

of GNN. Intuitively, the GNNEXPLAINER tries to find the minimum sub-graph $G_s \subset G$, *i.e.*, the minimum set of edges and nodes, hence enforcing explanation *sparsity* such that the model prediction is retained, *i.e.*, while ensuring explanation *fidelity*. The inferred sub-graph G_s is then regarded as the *explanation* for the graph G.

Formally, we aim to find a sub-graph $G_s = (V_s, E_s, H_s) \subset G$ such that the mutual information between the original prediction $y_{\max}$ and the sub-graph G_s is maximized, *i.e.*,

$$\max_{G_s} \mathrm{MI}(\hat{Y}, G_s) = \mathscr{H}(\hat{Y}) - \mathscr{H}(\hat{Y}|G = G_s) \tag{3.9}$$

which is equivalent to minimizing the conditional entropy:

$$\min_{G_s} \mathscr{H}(\hat{Y}|G = G_s) = -\mathbb{E}_{\hat{Y}|G_s}[\log(P_{\mathscr{M}}(\hat{Y}|G_s))] \tag{3.10}$$

Intuitively, we seek to extract the sub-graph G_s that maximizes the probability of $y_{\max}$. Exhaustively searching G_s in the space created by nodes V and edges E is infeasible due to the combinatorial nature of the task. Instead, GNNEXPLAINER formulates the task as an optimization problem that learns a mask to activate or deactivate parts of the graph. In this regard, this approach can be seen as a feature attribution method with *binarized* node and edge importance scores, *i.e.*, a node $v \in V$, edge $(u, v) \in E$, has importance one if $v \in V_s$, $(u, v) \in E_s$, respectively, and zero otherwise.

GNNEXPLAINER for node classification

The initial formulation by Ying et al. (2019) was developed for explaining *node classifiers*, where we wish to explain the classification prediction of a query node. Specifically, a mask $M_E \in \mathbb{R}^{|V|\times|V|}$ is learned over the edges, *i.e.*, over the adjacency matrix A. Masking edges will cut connections between the query node we wish to explain and its neighbors. Formally, we search for the mask such that:

$$\min_{M_E} -\sum_{c=1}^{C} \mathbb{1}_{[y=c]} \log(P_{\mathscr{M}}(\hat{Y}|G = A \odot \sigma(M_E), H))) \tag{3.11}$$

where C denotes the number of classes, σ is the sigmoid activation, and $\odot$ denotes element-wise multiplication. Heuristically, these constraints can be enforced by minimizing:

$$\mathscr{L} = \mathscr{L}_{\mathrm{KD}}(y_{\max}, y^{(l)}) + \alpha_{M_E} \sum_{i}^{|E|} \sigma(M_{E_i}^{(l)}) + \alpha_{\mathscr{H}} \mathscr{H}^e(\sigma(M_E^{(l)})) \tag{3.12}$$

where, l denotes the optimization step. The first term is a knowledge-distillation loss $\mathscr{L}_{\mathrm{KD}}$ between the new logits $y^{(l)}$ and the original prediction $y_{\max}$, ensuring explainer *fidelity*. The second term enforces explainer *sparsity* by minimizing the mask size M_E. The third term binarizes M_E by minimizing its element-wise entropy $\mathscr{H}^e$. Following Hinton et al. (2015),

$\mathscr{L}_{\text{KD}}$ is defined as a combination of distillation and cross-entropy loss:

$$\mathscr{L}_{\text{KD}} = \lambda \mathscr{L}_{\text{CE}} + (1-\lambda)\mathscr{L}_{\text{dist}} \text{ where } \lambda = \frac{\mathscr{H}^e(y^{(l)})}{\mathscr{H}^e(y_{\max})} \tag{3.13}$$

As the element-wise entropy $\mathscr{H}^e(y^{(l)})$ increases, $\mathscr{L}_{\text{CE}}$ gains importance and avoids predicting a different label. M_E, produced by optimizing Equation (3.12), is learned with iterative gradient descent until convergence is reached. An overview of GNNEXPLAINER optimization process is highlighted in Figure 3.1.

Note that the original formulation can be extended to prune features along the node dimension as well. As this extra step is not relevant for the proposed downstream tasks, we let the reader refer to Section 2.1 in Ying et al. (2019) for an in-depth formulation.

GNNEXPLAINER for graph classification

In order to be used for graph classification tasks, GNNEXPLAINER needs to be adapted. Indeed, the READOUT step in a graph classification GNN pools all the nodes to derive a graph-level representation. Therefore, even in the extreme case where all the edges are masked, the explanation would still include all the nodes, which does not fulfil the *sparsity desiderata*. The original paper proposes to use the largest connected sub-component induced by the masked graph as the explanation. However, this is a strong assumption and in many applications, the optimal explanation will be a disconnected graph. Furthermore, it is common that the nodes offer better *units of explanations* than edges, as they are often more intuitive and substantial information, *e.g.*, atoms are more informative than chemical bonds for molecular property prediction. To address these limitations, we propose to learn a mask that will directly operate at *node-level* .

Formally, we aim to learn a mask M_V that satisfies:

$$\min_{M_V} -\sum_{c=1}^{C} \mathbb{1}_{[y=c]} \log(P_{\mathscr{M}}(\hat{Y}|G=A, \sigma(\text{diag}(M_V))H))) \tag{3.14}$$

where $\text{diag} : \mathbb{R}^{|V|} \rightarrow \mathbb{R}^{|V|\times|V|}$ is the diagonal matrix of the weight vector M_V. As before, we intend the explanations to be as compact as possible, with binarized weights, while providing the same prediction as the original graph, which can be formulated as:

$$\mathscr{L} = \mathscr{L}_{\text{KD}}(y_{\max}, y^{(l)}) + \alpha_{M_V} \sum_{i}^{|V|} \sigma(M_{V_i}^{(l)}) + \alpha_{\mathscr{H}} \mathscr{H}^e(\sigma(M_V^{(l)})) \tag{3.15}$$

As for the node classification setting, M_V is learned by gradient descent. After convergence, the mask weights define the node-level importance scores. A thresholding is further applied

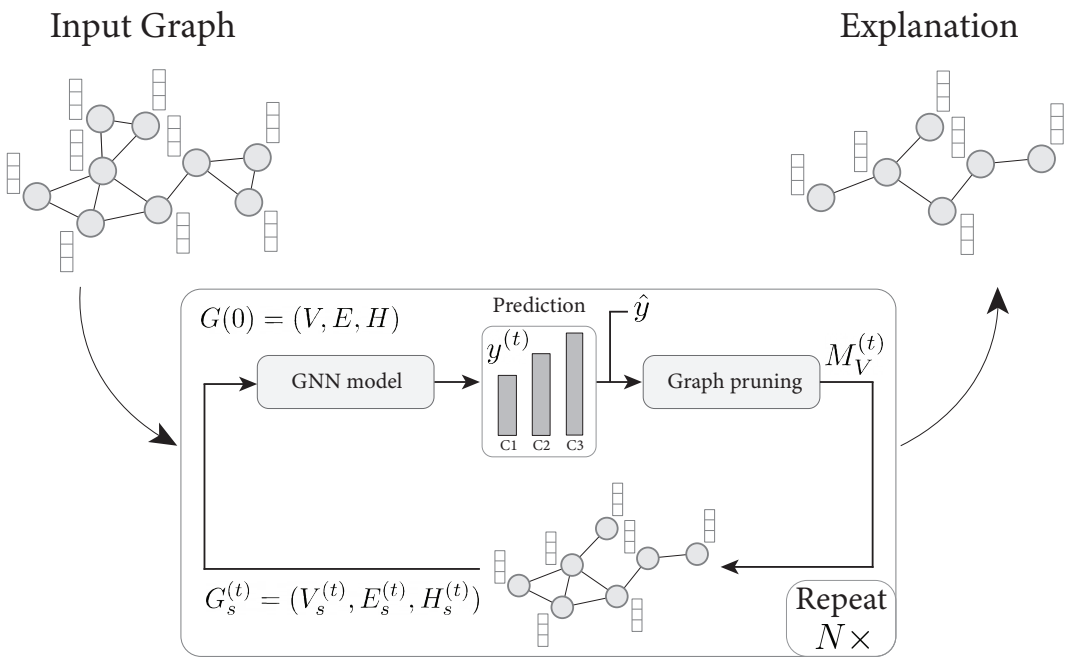


Figure 3.1 – Overview of GNNEXPLAINER iterative node pruning. A query graph is passed through a pre-trained GNN model where the graph prediction is stored. A node-level mask is learned to update the graph, *i.e.*, by deactivating some nodes, using the original label and the current prediction, the mask size, and the mask entropy. The masked graph is then re-processed by the GNN model for another iteration. The process is repeated until convergence.

on the node weights to extract only the most important ones.

3.4 A glimpse into qualitative results

This chapter focuses on the methodological aspects of deep graph interpretability. Applications of these methods will be provided in Chapter 6. Figure 3.2 provides node-level importance scores obtained to explain the prediction of a benign tumor region in an H&E histology image. This example is there to give the reader an intuition of how graph explainers work. An in-depth analysis will be conducted in Chapter 6.

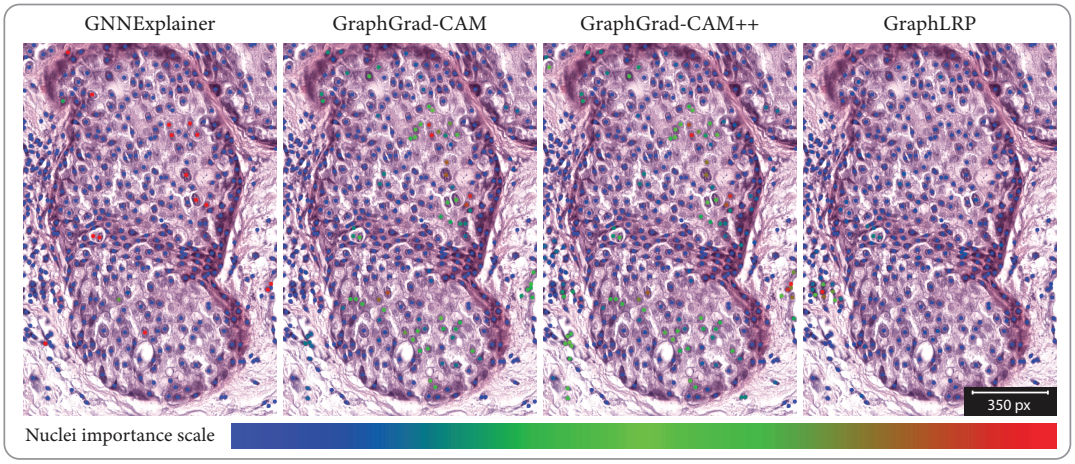


Figure 3.2 – Examples of post-hoc feature attribution methods to explain a Benign histology image. Important nodes are marked in red, and least important ones in blue.

Part II
Graph Representation and Modeling in Computational Pathology

4 Computational Pathology Background

In this chapter, we present basic concepts related to pathology, digital pathology and computational pathology. We begin by introducing pathological background required to understand the clinical relevance of the main contributions of this thesis. Next, we introduce DigPath and its potential to transform the way pathology is practiced. Finally, CompPath notions like stain normalization and entity-graph representations are presented. A reader familiar with CompPath can dispense with reading this chapter.

4.1 Pathology prerequisites

Pathology refers to the understanding of the causes and effects of diseases, based primarily on the analysis of tissue, cell and body fluid samples. When the study of biological tissues is performed at a microscopic level, that is, the study of microscopic anatomy, it is referred to as *histology*. Specifically, the examination of tissue biopsies or surgical specimens by a pathologist for medical diagnosis is referred to as *histopathology*. All the data used in this thesis are histopathology data acquired for cancer diagnosis and tumor detection. Tissue processing follows three steps. Namely, tissue specimen acquisition for tissue sample extraction, tissue specimen preparation in objective to highlight certain biomarkers, and tissue analysis for patient diagnosis and prognosis (see Figure 4.1).

4.1.1 Tissue specimen acquisition

Tissue specimen acquisition, or *biopsy*, involves extraction of tissue samples for examination to determine the presence and extent of a tumor. Depending on the organ to analyse and complementary diagnosis information, different biopsy types are performed. Fine needle aspiration biopsy are used to remove small samples of cells, for example if swellings or lumps were detected just under the skin. When larger tissue regions need to be extracted, core needle biopsies are employed. They use wider needles allowing for larger tissue sample extraction. This commonly employed technique is used for finding abnormalities, *e.g.*, detection of

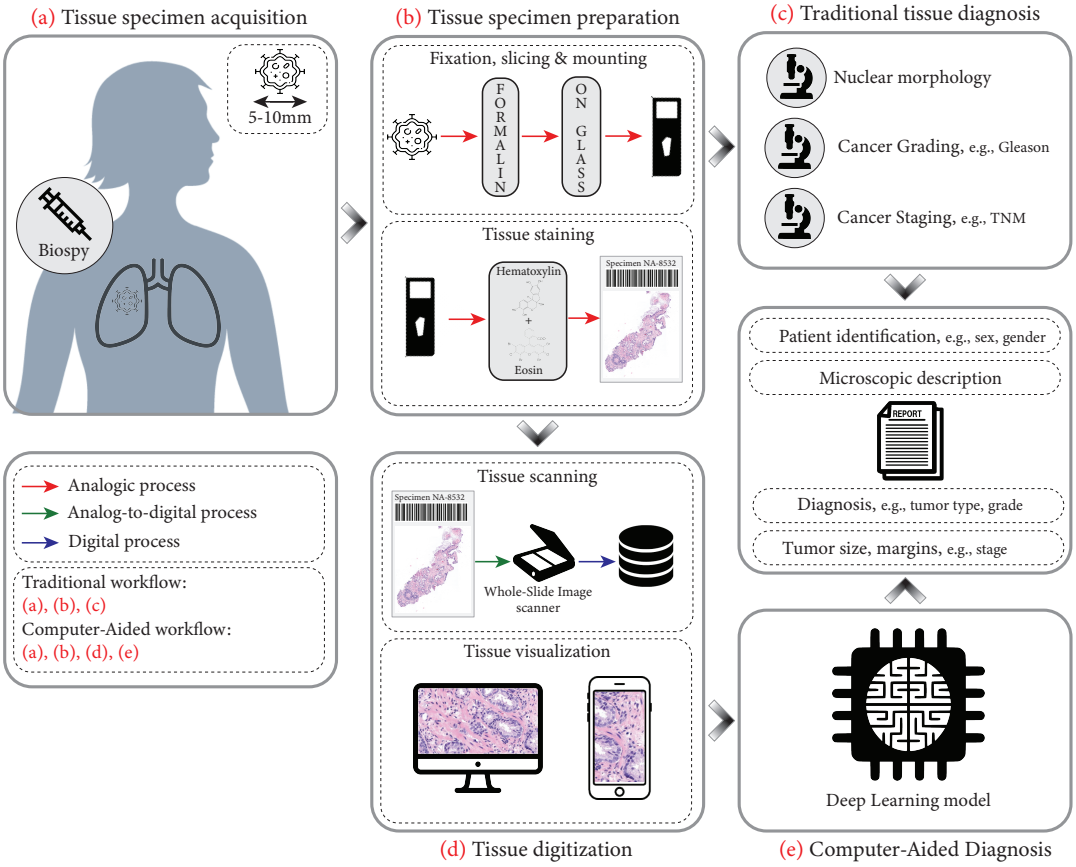


Figure 4.1 – Overview of a traditional ((a), (b), (c)) and AI-assisted ((a), (b), (d), (e)) diagnosis workflow. In (a), a tissue specimen is extracted with a biopsy. In (b), the tissue is prepared for microscopic analysis, including tissue fixation, thin slicing and mounting on a glass slide. In (c), a pathologist is conducting a diagnosis by analysing the tissue morphology towards grading and stating. Alternatively, in (d), the tissue is scanned to render a WSI, before being processed in (e) with a CAD tool.

malignant patterns, and to test for the presence of biomarkers, *e.g.*, hormone receptor status (ER, PR) in breast cancer. When tumorous regions have been detected and require surgery, surgical biopsies are employed, either to remove part of an abnormal tissue region (incisional biopsy), or to remove an entire abnormal region (excisional biopsy). In complement to the aforementioned techniques, sentinel lymph node biopsies can be required to find out if the tumor has spread to the nearest axillary lymph nodes.

4.1.2 Tissue specimen preparation

Before analysis, the extracted tissue samples need to be prepared. First, the tissue specimen is cut into thin sections, then it is mounted on a glass slide and stained with dyes before examination under a microscope.

As fresh tissue samples can be easily distorted and damaged, they need to be chemically preserved and fixed. Two methods exist to ensure the tissue is firm enough to be cut into thin sections: permanent paraffin-embedded sections and frozen sections. Permanent sections are prepared by placing the tissue in fixative, *e.g.*, in formalin, and then further processed with additional task-dependent solutions. The tissue is then placed in paraffin wax, in order to be cut into thin slices, which are then placed on glass slides for staining. This whole process is time-consuming and typically takes several days. Frozen sections are prepared by simply freezing the tissue sample before slicing it. The process takes about 15 to 20 minutes, and can be performed while a patient is in the operating room. Frozen sections are employed when an immediate answer is needed, *e.g.*, for tumor margin detection. While frozen sectioning is a much faster process, permanent sections remain the preferred option as they provide better quality for examination by pathologists.

In order for pathologists to visualize tissue morphology, *i.e.*, the tissue structure, tissue samples need to be stained. The gold-standard staining protocol is based on a combination of haematoxylin (H) dye to stain cell nuclei in blue and eosin (E) dye to stain extra-cellular structures, *e.g.*, stromal region, in pink and red. Hematoxylin & Eosin (H&E) staining allows for high-quality visualization of tissue structure, and detection of abnormal and cancerous nuclei. Even if H&E routine staining outcome forms an essential part of the diagnostic procedure, it is often complemented with other stainings for biomarker-specific analysis, *e.g.*, if H&E nuclei organisation is representative of two tumor subtypes that need to be discriminated. Complementary stainings include immunostaining based on immunohistochemistry, *e.g.*, HER2 protein detection and estrogen/progesterone receptors (ER/PR) status for breast cancer characterization.

4.1.3 Tissue analysis and diagnosis

In a clinical setting, pathologists begin the analysis of a tissue biopsy by discerning the morphology and the spatial distribution of tissue parts, such as epithelium, stroma, necrosis, etc. Then, they localize their analysis to specific tissue regions to evaluate nuclear phenotype, morphology, topology, and tissue distribution among several other criteria for the classification. In particular, pathologists examine samples to determine the tumor *grade* and tumor *stage*.

The purpose of tumor grading is to determine the appearance of abnormal tumor cells. It is a prognosis marker for the rate at which the tumor can grow and spread. If the cells are close to normal, they are referred to as *well-differentiated*, and correspond to slow cancer growth and spread. If they look abnormal, they are said to be *undifferentiated* or *poorly differentiated*. There exist different grading systems that are cancer- and organ-specific. For instance invasive breast cancer can be graded with the Nottingham system (Rakha et al., 2008), which is based on three observable features: (i) tubule formation (on a scale from 1 to 3) that describes the amount of gland formation, (ii) the nuclear grade (1 to 3 scaling) that evaluates nuclei pleomorphism, *i.e.*, the size and shape of cancerous nuclei in tumor cells, and (iii) the mitotic

rate (1 to 3 scaling) that represents how much the tumor cells are proliferating. By simply summing up the score for each feature, we derive the final grade as: Grade 1 (low grade / well differentiated) if score in [3,5], Grade 2 (Intermediate grade / moderately differentiated) if score in [6,7], and Grade 3 (High grade or poorly differentiated) if score > 7. Differently, prostate cancer is graded with the Gleason grading system (Chen et al., 2008), a 6 to 10 scale that indicates cell differentiation, from well differentiated to anaplastic. Specifically, the Gleason grade is derived by considering the two largest most aggressive cancer patterns in the observed tissue sample, where each pattern can be of type benign, grade 3, 4, or 5.

The cancer grade is complemented by the cancer stage that describes the tumor size and if the tumor has spread outside its origin organ. Most cancer types are staged using the TNM system, and are based on three factors: Tumor (T), Node (N), and Metastasis (M). T quantifies the primary tumor size on a 0 to 4 scale, where 0 denotes no cancer, and 1,2,3,4 larger cancer regions. N indicates if the tumor has spread to the lymph nodes, and if yes it encodes the number of affected axillary lymph nodes. Finally, M encodes if the cancer has spread to other body parts. By summing up individual scores, the cancer stage is derived on a scale from 0 (non-invasive cancer) to 4 (metastatic cancer).

The cancer grade, stage, or specific biomarkers can be used as training targets for developing Computer-Aided Diagnosis (CAD) tools, *e.g.*, Gleason grade prediction from core needle biopsies as presented in Chapter 7.

4.2 Prerequisites in Computational Pathology

Computational pathology refers to the use of *computational* methods to automate pathology tasks. Developing computational approaches can be used for computer-aided diagnosis (Van der Laak et al., 2021; Campanella et al., 2019; Lu et al., 2021b; Litjens et al., 2017; Deng et al., 2020) or discovering new cancer biomarkers (Lu et al., 2021a; Gamble et al., 2021). A CompPath pipeline for CompPath is exemplified in Figure 4.1.

4.2.1 Digital Pathology

Recently, new medical imaging techniques have been developed to scan tissue slides, allowing to transform a tissue specimen into a high-resolution image, called a WSI. Scanners have consistently evolved in the past twenty years to today being able to digitize a slide in less than 30 seconds at multiple magnifications. WSI are typically giga-pixel images, *e.g.*, a breast biopsy can be as large as $100'000 \times 100'000$ pixels at $40\times$ magnification. In 2017, the Federal Drug Administration (FDA) has approved the first WSI system for routine diagnostic practice (Boyce et al., 2017). Since, more and more labs are modernizing their infrastructure with digital pipelines. Digitizing the clinical workflow offers several advantages. First, digitized slides allow for efficient and scalable storage, and eliminate the need for costly and time-consuming physical archiving. It also enables the development of centralized queryable databases that

can be accessed from anywhere, for instance to easily compare slides. It also facilitates communication between pathologists, oncologists and radiologists, *e.g.*, to ask confirmation and complementary information when working with challenging cases. The development of friendly User Interface (UI) where pathologists can easily annotate diagnostically relevant regions, *e.g.*, for counting tasks, helps standardization and reproducibility. Pathology reports also greatly benefits from digitization, by ensuring the use of standardized templates that can automatically be added to hospital databases. Finally, and this is the focus of this thesis, a digitized environment enables the integration of computational methods in the workflow, *e.g.*, as being part of a UI or running cases in the background for enhanced diagnosis.

4.2.2 Tissue preprocessing and stain normalization

Due to the lack of standardized and automated staining procedures, stained tissue images exhibit appearance variability, *e.g.*, explained by different specimen preparation techniques, staining protocols, fixation characteristics, different data acquisition methods (scanners, digitization artifacts, etc.). Such variability adversely impacts computational methods for downstream diagnosis (Veta et al., 2014; Tellez et al., 2019b). To alleviate the stain variability, a stain normalization algorithm is often employed. In the context of H&E staining, two popular techniques have been proposed by Macenko et al. (2009) and Vahadane et al. (2016).

Macenko stain normalization

Macenko stain normalization (Macenko et al., 2009) algorithm is based on the principle that the RGB colors of each pixel is a linear combination of two unknown stain vectors, Hematoxylin and Eosin, that need to be estimated. First, the algorithm estimates the stain vectors of a H&E image by using a Singular Value Decomposition (SVD) of non-background pixels. Second, the algorithm applies a correction to account for the intensity variations due to noise. The algorithm requiring no model training is computationally inexpensive. In practice, the scalable and fast pipeline proposed by Stanisavljevic et al. (2018) is often used.

Vahadane stain normalization

Vahadane stain normalization (Vahadane et al., 2016) builds on Macenko et al. (2009) work to propose a solution for both stain separation and color normalization. Specifically, stain vectors are evaluated by applying sparse non-negative matrix factorization of a reference image. While providing a slightly better rendering, this approach is computationally more expensive (see Appendix B).

4.3 Graphs in Computational Pathology

Graphs in CompPath are proposed to realize the tissue composition-to-functionality relationship in terms of the phenotypical and structural characteristics of the tissue. An overview of the use of graph representation for CompPath is presented in Table 4.1 and a comprehensive review is proposed in Ahmedt et al. (2021). The motivation justifying the pertinence of using graph representations and graph learning methods for modeling histology images will be detailed in the subsequent chapters. In the sequel, we provide complementary preliminary notions for working with graphs in CompPath.

Cell-graphs: We define a cell-graph (see Figure 4.2(a)) as a graph representation of an histology image, where the nodes depict nuclei, and edges encode nuclei-nuclei interactions. Cell-graph representations were initially proposed by Demir et al. (2004) and were since used for addressing various pathology tasks, including cancer grading, cancer stratification, neural network interpretability, among others (Demir et al., 2004; Zhou et al., 2019a; Wang et al., 2020; Chen et al., 2020; Pati et al., 2021a; Jaume et al., 2021b).

Tissue-graphs: We define a tissue-graph (see Figure 4.2(b)) as a graph-structured representation where nodes represent tissue components and edges tissue-tissue interactions. Tissue-graphs can efficiently model large images by encoding consistently morphological regions as graph nodes (Pati et al., 2021a; Zheng et al., 2019; Anklin et al., 2021; Lu et al., 2021a).

We refer to cell- and tissue-graphs as *biological entity-graphs*, as the nodes are biologically defined and correspond to diagnostically relevant entities that pathologists can relate to and reason with.

Patch-graphs: As the name suggests, patch-graphs (Anand et al., 2020; Adnan et al., 2020; Aygunes et al., 2020; Zhao et al., 2020; Li et al., 2018a; Levy et al., 2021) are built by defining patches as nodes and patch-to-patch relations as edges. Patch-graphs offer easy-to-use representations for encoding large histology images (see Figure 4.2(c)).

Hierarchical-graphs: Uni-level graphs can be combined to form hierarchical graphs. For instance, low-level cell information can be encoded in a cell-graph, intermediate-level information in a tissue-graph and high-level information in a gland-graph (a graph where glands represent nodes and glandular interactions encode edges).

All the aforementioned graph types would typically be associated with node features, *e.g.*, handcrafted or DL features, to characterize the entities. The topology can depict the spatial or semantic relationship among the entities, *e.g.*, k-Nearest Neighbors (k-NN), region adjacency, or probabilistic models. The graphs can be processed using classic Machine Learning (ML) (Sharma et al., 2016, 2017) or GNNs that proved to outperform state-of-the-art CNN-based approaches for several pathology tasks across multiple organs (Garciá-Arteaga et al., 2017; Zhou et al., 2019a; Zhao et al., 2020; Adnan et al., 2020; Pati et al., 2021a; Studer et al., 2021; Anklin et al., 2021).

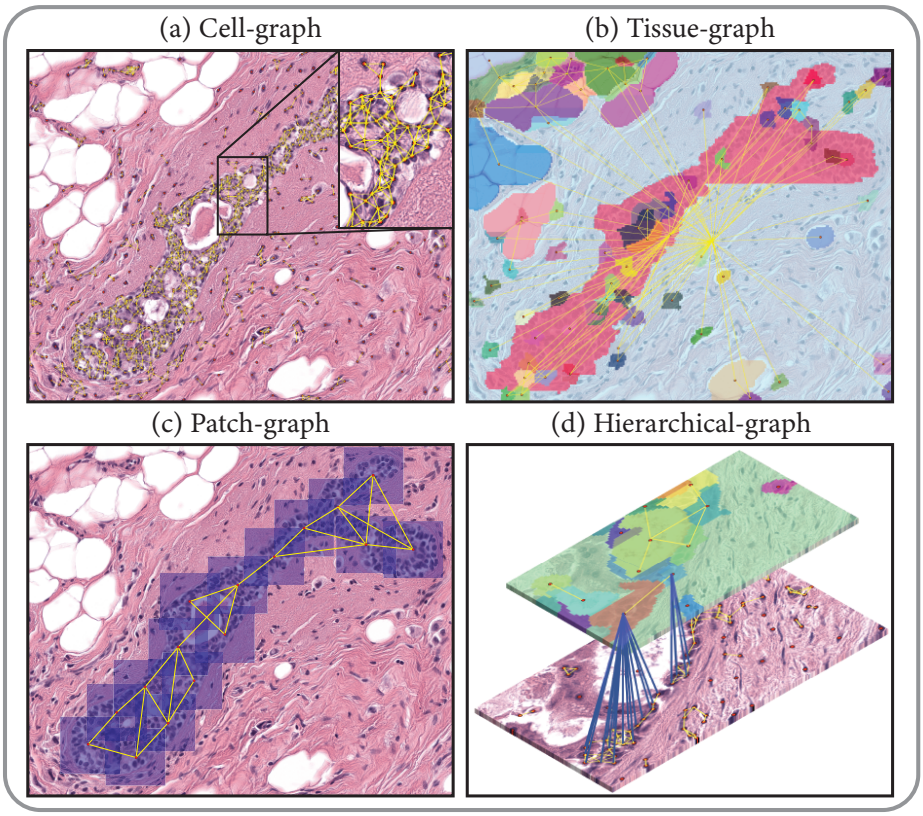


Figure 4.2 – Examples of graph-based representations of histology images. Nodes can encode biological entities, *e.g.*, (a) nuclei in cell-graphs, (b) tissue components in tissue-graphs or (c) patches in patch-graphs. (d) Graph representations can be hierarchical to encode tissue composition in the form of a hierarchical-graph, *e.g.*, by encoding a cell-graph, tissue-graph and cell-to-tissue connections.

	Reference	Task	Organ	Modality	Image size	Dataset size
Cell-graph	Zhou et al. (2019a)	Cls.	Breast	TRoI	$4,548 \times 7,520$	139
	Wang et al. (2020)	Cls.	Prostate	TMA	-	886
	Anand et al. (2020)	Cls.	Breast	TRoI	$2,048 \times 1,536$	400
	Sureka et al. (2020)	Cls.	Breast	TRoI	2048×1536	400
		Cls.	Prostate	TMA	$3,100 \times 3,100$	1,506
	Jaume et al. (2020)	Cls.	Breast	TRoI	$2000 \times 2,000$	2,080
	Jaume et al. (2021b)	Cls.	Breast	TRoI	$1,900 \times 1,900$	4,391
	Studer et al. (2021)	Cls.	Colon	TRoI	-	520
Tissue-graph	Zheng et al. (2019)	Cls.	Breast	WSI	-	150
	Lu et al. (2020)	Cls.	Breast	WSI	-	709
	Anklin et al. (2021)	Seg.	Prostate	TMA	3100×3100	1,506
		Seg.	Prostate	WSI	$11,000 \times 3,000$	155
	Jaume et al. (2021c)	Seg.	Prostate	WSI	11000×3000	155
		Seg.	Prostate	WSI	$11,000 \times 11,000$	5,759
		Seg.	Prostate	WSI	$11,000 \times 11,000$	5,662
Patch-graph	Li et al. (2018a)	Cls.	Lung	WSI	-	535
		Cls.	Lung	WSI	-	491
		Cls.	Lung	WSI	-	425
	Wu et al. (2019)	Cls.	Skin	WSI	-	1241
	Ozen et al. (2020)	Cls.	Breast	TRoI	-	1,080
	Aygunes et al. (2020)	Cls.	Breast	TRoI	-	1,030
	Zhao et al. (2020)	Cls.	Colon	WSI	-	425
	Raju et al. (2020)	Cls.	Colon	WSI	-	1,345
	Ding et al. (2020)	Cls.	Colon	WSI	-	421
	Adnan et al. (2020)	Cls.	Lung	WSI	-	1,026
Hierarchical-graph	**Pati et al. (2020)**	Cls.	Breast	TRoI	$2,000 \times 2,000$	2,080
	Zhang et al. (2020)	Cls.	Breast	TRoI	$2,048 \times 1,536$	400
	Chen et al. (2020)	Cls.	Renal	TRoI	-	-
		Cls.	Brain	TRoI	-	-
	Shi et al. (2020)	Cls.	Cervical	TRoI	-	4,039
		Cls.	Cervical	TRoI	128×128	25,378
	Pati et al. (2021a)	Cls.	Breast	TRoI	$1,900 \times 1,900$	4,391
	Levy et al. (2021)	Reg.	Colon	WSI	-	172
		Reg.	Lymphoma	WSI	-	84
Misc	**Jaume et al. (2021a)**	Helpers	Agnostic	-	-	-
	Ahmedt et al. (2021)	Review	-	-	-	-

Table 4.1 – Overview of graph representations and models in CompPath, grouped by graph type: cell-graphs, tissue-graphs, patch-graphs and hierarchical-graphs. Publications included in this thesis are highlighted in **bold**. Cls., Reg., Seg. stands for a classification, regression and segmentation, respectively.

5 Hierarchical Graph Representations of Histology Images

The ideas, methods and results presented in this chapter are published in:

- "Hierarchical Graph Representations in Digital Pathology", Pushpak Pati*, **Guillaume Jaume***, Antonio Foncubierta, Florinda Feroce, Anna Maria Anniciello, Giosuè Scognamiglio, Nadia Brancati, Maryse Fiche, Estelle Dubruc, Daniel Riccio, Maurizio Di Bonito, Giuseppe De Pietro, Gerardo Botti, Jean-Philippe Thiran, Maria Frucci, Orcun Goksel, Maria Gabrani. In *Medical Image Analysis*, 2021 (Pati et al., 2021a).

- "HACT-Net: A Hierarchical Cell-to-Tissue Graph Neural Network for Histopathological Image Classification", Pushpak Pati*, **Guillaume Jaume***, Lauren Alisha Fernandes, Antonio Foncubierta-Rodríguez, Florinda Feroce, Anna Maria Anniciello, Giosue Scognamiglio, Nadia Brancati, Daniel Riccio, Maurizio Di Bonito, Giuseppe De Pietro, Gerardo Botti, Orcun Goksel, Jean-Philippe Thiran, Maria Frucci, Maria Gabrani. In *International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI), MICCAI Workshop on Graphs in Medical Imaging*, 2020 (Pati et al., 2020).

GJ (the author of this thesis) is sharing first co-authorship with PP. The ideas, concepts and experiments were designed by GJ and PP. GJ was responsible for implementing the graph neural network models, some of the image-to-graph modules and an experiment manager. GJ executed the experiments with the help of PP and LAF. FF, AMA, GS, MF, ED, MDB generated and annotated the BRACS data. They also served as a medical guarantee to justify the problem statement, ensure the use of appropriate metrics, and for the evaluation of the AI models. AF, NB, DR, MDB, GDP, GB and MG were responsible for coordination and management of the project. JPT and OG supervised and supported GJ in organizing his research. The manuscript was written by GJ and PP and subsequently revised by LAF, OG and MG.

5.1 Introduction

Deep learning techniques primarily use CNNs (Madabhushi and Lee, 2016; Parwani, 2019) to process histology images in a patch-wise manner. CNNs extract representative patterns from patches and aggregate them to perform image-level tasks, *e.g.*, tumor detection, tumor staging, tumor subtyping. However, patch-wise processing suffers from the trade-off between the resolution of operation and the utilization of adequate context (Bejnordi et al., 2017; Sirinukunwattana, 2018). Operating at a higher resolution captures local cellular information but limits the field-of-view due to computational burden and limits the access to global tissue microenvironment information. In contrast, operating at a lower resolution hinders resolvability of cells and access to cellular properties. Bejnordi et al. (2017); Sirinukunwattana (2018); Tellez et al. (2019a) have proposed CNN methods to address such trade-off by leveraging visual context, however, CNNs, which operate on fix-sized input patches, are confined to a fixed field-of-view and are restricted to incorporate information from varying spatial distances. Further, pixel-based processing in CNNs disregards the notion of histologically meaningful entities (Hagele et al., 2020), such as cells, glands, and tissue types. The inattention to histological entities severely limits the interpretability of CNNs by pathologists, and any utilization of established entity-level prior pathological knowledge in the CNN-based diagnostic frameworks. Additionally, CNNs disregard the structural composition of tissue, where fine entities hierarchically constitute to form coarser entities, such as, epithelial cells organize to form epithelium, which further constitutes to form glands. Such a hierarchical structure analysis is of high value for the diagnosis and prognosis.

In this chapter, we address the aforementioned limitations by shifting the analytical paradigm from pixel to entity-based processing. In an entity paradigm, a histology image is described as an entity-graph, where nodes and edges of a graph denote biological entities and inter-entity interactions, respectively. An entity-graph can be customized in various aspects, *e.g.*, in terms of the type of entity set, entity attributes, and graph topology, by incorporating any task-specific prior pathological knowledge. Thus, the graph representation enables pathology-specific interpretability and human-machine co-learning. In addition, the graph representation is memory efficient compared to images and can seamlessly describe a large tissue region. Demir et al. (2004) first introduced cell-graphs using cells as the entity type. Though a cell-graph efficiently encodes the cell microenvironment, it cannot extensively capture the tissue microenvironment, *i.e.*, the distribution of tissue regions such as necrosis, stroma, epithelium, etc. Similarly, a tissue-graph comprising of the set of tissue regions cannot depict the cell microenvironment. Therefore, an entity-graph representation using a single type of entity set is insufficient to comprehensively describe the tissue composition. To address this, we propose a multi-level entity-graph representation, *i.e.*, HierArchical Cell-to-Tissue (HACT), consisting of multiple types of entity sets, *i.e.*, cells and tissue regions, to encode both cell and tissue microenvironment. The multiset of entities is inherently coupled depicting tissue composition at multiple scales. The HACT graph encodes individual entity attributes and intra- and inter-entity relationships to hierarchically describe a histology image. Upon the graph construction,

a GNN processes the entity-graph to perform image analysis. Specifically, we introduce a hierarchical GNN, HierArchical Cell-to-Tissue Network (HACT-Net), to sequentially operate on the HACT graph, from fine-level to coarse-level, to provide a fixed dimensional embedding for the image. The embedding encodes morphological and topological distribution of the multiset of entities in the tissue. Interestingly, our proposed methodology resembles the tissue diagnostic procedure in clinical practice, where a pathologist hierarchically analyzes a tissue.

We propose a methodology that consists of HACT graph construction and HACT-Net based histology image analysis. We characterize breast Tumor Regions-of-Interest (TRoIs) to evaluate our methodology. Specifically, the contributions presented in this chapter are:

- A novel hierarchical entity-graph representation (HACT) and hierarchical learning (HACT-Net) methodology for analyzing histology images;
- Introducing a public dataset, BReAst Carcinoma Subtyping (BRACS[1]), a large cohort of breast TRoIs annotated with seven breast cancer subtypes. BRACS includes challenging atypical cases and a wide variety of TRoIs representing a realistic breast cancer analysis;
- An evaluation of our proposed methodology on the BRACS dataset where an extensive assessment demonstrates our classification performance outperforming several recent CNN and GNN approaches for cancer subtyping. On an independent study, HACT-Net even outperforms three independent pathologists on per-class and aggregated classification tasks.

5.2 Related work

5.2.1 Cancer subtyping

Several deep learning algorithms have been proposed to categorize histopathology images into cancer subtypes (Komura and Ishikawa, 2018; Srinidhi et al., 2021; Deng et al., 2020; Spanhol et al., 2016; Araujo et al., 2005; Aresta et al., 2019). For this task, most algorithms employ CNNs in a patch-wise manner: In Araujo et al. (2005); Bardou et al. (2018); Roy et al. (2019); Mercan et al. (2019a), CNNs are used to classify breast histology images. These methods use single stream patch-wise approaches to capture local patch-level context, aggregate the patch-level information, and classify the image using aggregated information. For instance, Mercan et al. (2019a) aggregate the patch-wise CNN generated embeddings and class-probabilities to construct a class-probability weighted TRoI-level feature representation, and subsequently classifies the TRoI. However, single-stream approaches do not capture adequate context from the tissue microenvironment to aptly encode a patch. Sirinukunwattana (2018) address this issue by including multi-scale information from concentric patches across different magnifications. Tellez et al. (2019a) propose neural image compression, where WSIs are compressed using a neural network trained in an unsupervised fashion, followed by a CNN trained on

[1] BRACS dataset for breast cancer subtyping: https://www.bracs.icar.cnr.it

the compressed representations to classify the images. Shaban et al. (2020) include an attention module with an auxiliary task to improve neural image compression for histology image classification. Yan et al. (2020) propose a hybrid convolutional and RNN to utilize spatial correlations among patches for analyzing histology images. Bejnordi et al. (2017) propose a stacked CNN architecture to capture large contexts and perform end-to-end processing of large histology images. Pinckaers et al. (2020) propose a streaming CNN to accommodate multi-megapixel images. Campanella et al. (2019) utilize a Multiple Instance Learning (MIL) approach to process whole-slide images in an end-to-end manner. Though the aforementioned methods use different strategies to encode a tissue, they all operate on a square and fix-sized patches. However, actual TRoIs can be of highly varying dimensions and shapes depending on the cancer subtype and the site of tissue extraction. In contrast, our proposed entity-graph methodology can acquire both local and global context from arbitrary-sized TRoIs.

The approaches followed by researchers to circumvent this limitation have been to either downsample the WSI image to a smaller resolution or dividing the WSI in many small patches and aggregating results later on. Cid et al. (2018) have shown that the tumor stroma environment correlates with prognosis, and patch-based techniques would not be able to capture this. On the other hand there are specific breast cancer subtypes like flat epithelial atypia (FEA) that are characterized by patterns that are best distinguished at larger resolutions as described in (Lerwill, 2008).

5.2.2 Graphs in computational pathology

Entity graph-based tissue representations can effectively describe the tissue composition by incorporating morphology, topology, and interactions among biologically comprehensible entities, unlike CNNs. Using cells as entities, Demir et al. (2004) introduced a cell-graph (CG) representation of a tissue, where cell morphology can be embedded in the nodes via handcrafted (Demir et al., 2004; Zhou et al., 2019a; Pati et al., 2020) or deep-learning based features (Chen et al., 2020). The graph topology is often heuristically defined, *e.g.*, using k-Nearest Neighbors, probabilistic modeling, or a Waxman model (Sharma et al., 2015). Subsequently, a CG is processed by classical machine learning techniques (Sharma et al., 2016, 2017) or GNNs (Zhou et al., 2019a; Pati et al., 2020; Chen et al., 2020; Anand et al., 2020) for mapping to tissue function. Recently, graph representations using patches (Aygunes et al., 2020) and tissue regions (Pati et al., 2020; Anklin et al., 2021) as entities have been proposed for better tissue representation. Other graph-based applications in computational pathology include cellular community detection (Javed et al., 2020), WSI classification (Zhao et al., 2020; Adnan et al., 2020), WSI segmentation (Anklin et al., 2021). Notably, entity-graphs consist of biological entities to which the pathologists can readily relate. So, the entity-graph paradigm enables to incorporate pathologically-defined, task-specific entity-level prior knowledge in constructing "meaningful" tissue representations. This implicitly enables *interpretability* and *explainability* of graph-based networks for pathologists as detailed in Chapter 3. For instance, Zhou et al.

(2019a) analyzes the clustering of nodes in a CG to group cells according to their appearance and tissue types. Sureka et al. (2020) employs robust spatial filtering that utilizes an attention-based GNN and node occlusion to highlight cell contributions.

5.3 Methodology

In this section, we detail our proposed methodology for hierarchical tissue analysis, as illustrated in Figure 5.1. For an input H&E stained histology TRoI image, first, we apply pre-processing to standardize the input. Then, we identify pathologically relevant entities and construct a HACT graph representation of the TRoI by incorporating the morphological and topological distribution of the entities. Finally, HACT-Net, a hierarchical GNN, is devised to map the HACT graph to a corresponding category, *e.g.*, cancer subtype.

5.3.1 Notation

Following the notation introduced in Chapter 1, we define an attributed entity-graph $G := (V, E, H)$ as a set of nodes V, edges E, and node features H. Each node $v \in V$ is represented by a feature vector $h(v) \in \mathbb{R}^d$, thus, $H \in \mathbb{R}^{|V| \times d}$. d denotes the number of features per node, and $|.|$ denotes set cardinality. The graph topology is described by an adjacency matrix $A \in \mathbb{R}^{|V| \times |V|}$, where $A_{u,v} = 1$ if $(u, v) \in E$. The neighborhood of a node $v \in V$ is denoted as $\mathcal{N}(v) := \{u \in V \mid v \in V, (u, v) \in E\}$.

5.3.2 Graph representation

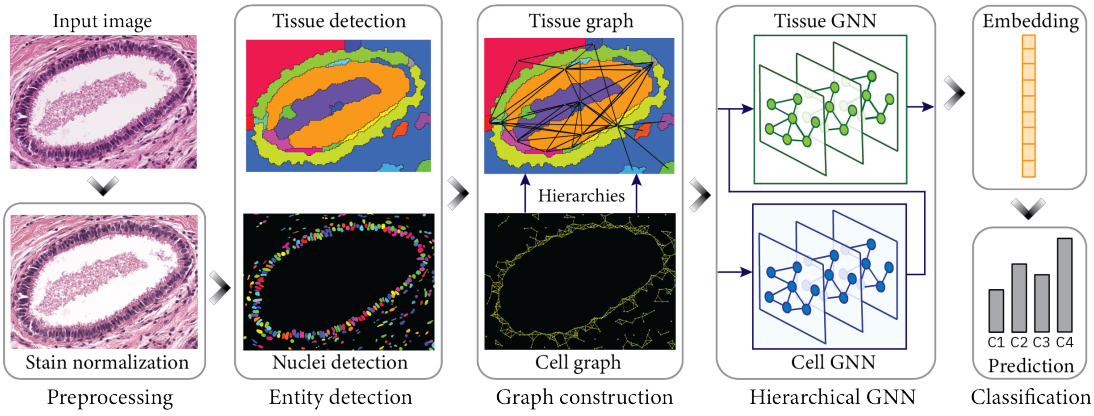


Figure 5.1 – Overview of the proposed hierarchical entity-graph based tissue analysis methodology. Following some pre-processing, a hierarchical entity-graph representation of a tissue is constructed, and it is processed via a hierarchical GNN to learn the mapping from tissue compositions to respective tissue categories. (Figure is best viewed in color.)

We first apply Macenko's stain normalization algorithm (see Chapter 4) to an input image in order to reduce stain variability across different samples. Then, the stain normalized TRoI is

processed to identify relevant entities and construct a hierarchical entity-graph representation. In this work, we consider nuclei and tissue regions as the entities. Therefore, the HACT graph consist of three components: 1) a low-level *cell-graph*, capturing cell morphology and interactions, 2) a high-level *tissue-graph*, capturing morphology and spatial distribution of tissue regions, and 3) cells-to-tissue hierarchies, encoding the relative spatial distribution of cells with respect to the tissue distribution. The details of the components are presented in the following subsections.

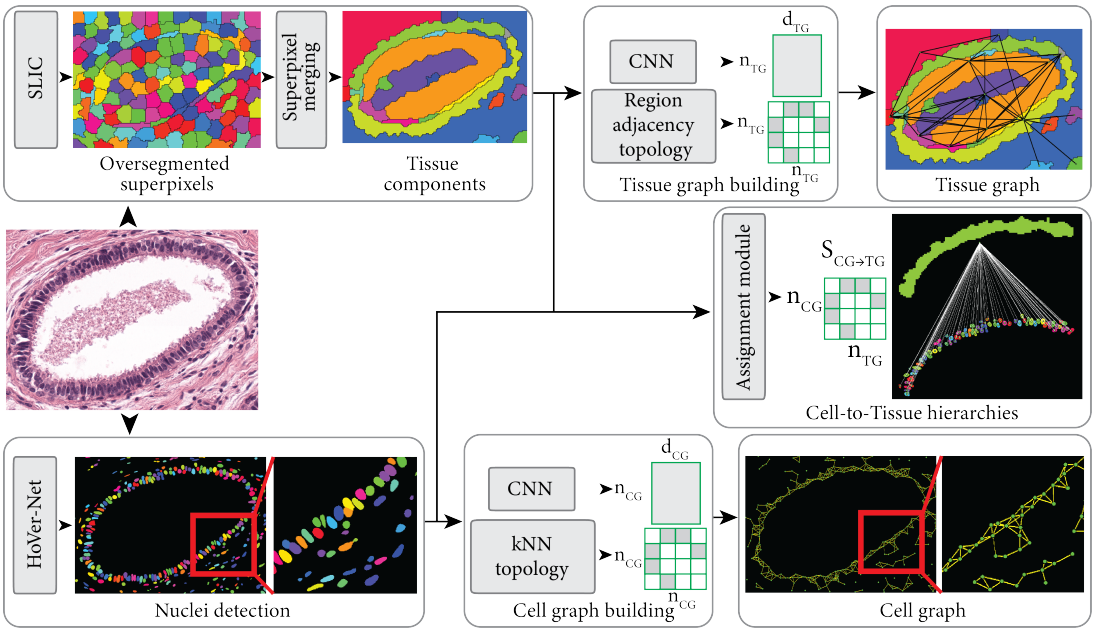


Figure 5.2 – Overview of hierarchical cell-to-tissue (HACT) graph construction for a TRoI. Our HACT graph representation consists of a cell-graph, a tissue-graph, and cell-to-tissue hierarchies, while encoding the phenotypical and topological distributions of tissue entities to describe the cell and tissue microenvironments. (Figure is best viewed in color.)

Cell-graph representation

A cell-graph (CG) characterizes the cell microenvironment, where nodes denote cells and encode cellular morphology, and edges denote cellular interactions and encode cellular topology. It is constructed in three steps, i) nuclei detection, ii) nuclei feature extraction, and iii) topology configuration, as shown in Figure 5.2.

Precise nuclei detection enables reliable CG representation. To this end, we use HoVer-Net, a nuclei segmentation network proposed by (Graham et al., 2019a), pre-trained on MoNuSeg dataset (Kumar et al., 2017). HoVer-Net leverages the instance-level information encoded in the vertical and horizontal distances of nuclear pixels to their centers of mass. These distances are used to accurately segment clustered nuclei, particularly in areas with overlapping nuclei. The centroids of the segmented nuclei form the spatial coordinates of nodes in the CG.

Following nuclei detection, morphological features are extracted by processing patches of size $h \times w$ centered around nuclei centroids with a ResNet network (He et al., 2016) that was pre-trained on the ImageNet dataset (Deng et al., 2009a). Spatial features of the nuclei are extracted as the spatial coordinates of the nuclei, normalized by the TRoI dimensions. Morphological and spatial features together constitute the nuclei features, which are collocated for all nodes as the node-feature matrix $H_{\text{CG}} \in \mathbb{R}^{|V_{\text{CG}}| \times d_{\text{CG}}}$.

The CG topology E_{CG} is based on the fact that spatially close cells have stronger interactions (Francis and Palsson, 1997) while distant cells having weaker cellular interactions. Accordingly, we connect nearby cells with edges to model their interactions. To this end, we use the k-NN algorithm to build an initial topology, that we subsequently prune by removing edges longer than a threshold distance d_{min}. We use the Euclidean distances between nuclei centroids in the image space to quantify cellular distances. The resulting CG topology is represented by a binary adjacency matrix $E_{\text{CG}} \in \mathbb{R}^{|V_{\text{CG}}| \times |V_{\text{CG}}|}$. Figure 5.2 illustrates the CG representation for a sample TRoI. Formally, a CG representation is formulated as $G_{\text{CG}} := \{V_{\text{CG}}, E_{\text{CG}}, H_{\text{CG}}\}$.

Tissue-graph representation

A tissue-graph (TG) depicts a high-level tissue microenvironment, where the nodes and edges denote tissue regions and their interactions, respectively. A TG is constructed by first identifying tissue regions (*e.g.*, epithelium, stroma, lumen, necrosis etc.), followed by encoding the tissue regions, and finally the topology building. The steps are illustrated in Figure 5.2. A parallel approach involving superpixel detection and neighborhood information aggregation is adopted by (Mercan et al., 2018) to semantically segment tissue regions in histology images.

Tissue regions are identified in two steps. First, we oversegment the tissue to detect non-overlapping homogeneous superpixels. We operate at a low magnification to avoid noisy pixels and reduce computational cost. Specifically, we use the Simple Linear Iterative Clustering (SLIC) algorithm (Achanta et al., 2012). SLIC follows an unsupervised approach by associating each pixel with a feature vector and merging the pixels using a localized version of k-means clustering. Next, we iteratively merge neighboring superpixels that have similar color attributes, *i.e.*, channel-wise mean, to create superpixels that capture meaningful tissue information. A sample tissue-region instance-map is shown in Figure 5.2.

To extract feature representations of tissue regions, we follow a two-step procedure: first, we extract CNN-based features for oversegmented superpixels, *i.e.*, patches of size $h \times w$ centered around the superpixel centroids are processed by ResNet. Second, morphological features of a tissue region are obtained by averaging the deep features of its constituting superpixels. Similar to CG, we include spatial features as the normalized centroids of the tissue region. For a TRoI with a set of V_{TG} tissue regions, we denote the TG node-feature matrix as $H_{\text{TG}} \in \mathbb{R}^{|V_{\text{TG}}| \times d_{\text{TG}}}$.

We assume adjacent tissue regions to biologically interact the most, and thus connect in

the TG topology. To this end, we construct a Region Adjacency Graph (RAG) (Potjer, 1996) where an edge is built between adjacent tissue regions. The topology is presented by a binary adjacency matrix $E_{\text{TG}} \in \mathbb{R}^{|V_{\text{TG}}| \times |V_{\text{TG}}|}$. Formally, a TG representation is formulated as $G_{\text{TG}} := \{V_{\text{TG}}, E_{\text{TG}}, H_{\text{TG}}\}$.

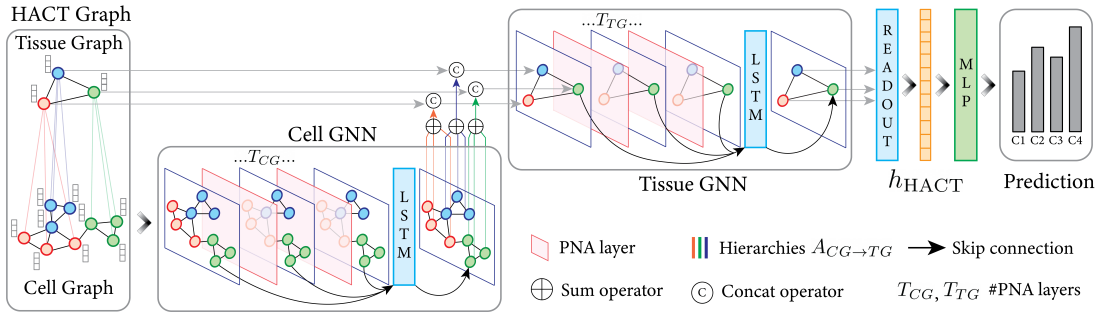


Figure 5.3 – Overview of the proposed HACT-Net architecture. The network processes an input HACT graph representation in a hierarchical manner, from fine cell-level to coarse tissue-region level, to obtain a contextualized graph embedding, and consequently classify the input graph. (Figure is best viewed in color)

Hierarchical Cell-to-Tissue graph representation

Tissues in histopathology can be viewed as a hierarchical organizations of biological entities ranging from fine-level, *i.e.*, cells, to coarse-level, *i.e.*, tissue regions. There exist intra- and inter-level coupling based on topological distributions and interactions among the entities. Following this motivation, we propose HACT, a HierArchical Cell-to-Tissue (HACT) graph representation to jointly represent low-level CG and high-level TG. Intra-level topology is already captured by the cell- and tissue-graphs. Inter-level topology is presented by a binary assignment (cell-to-tissue hierarchy) matrix $A_{\text{CG}\to\text{TG}} \in \mathbb{R}^{|V_{\text{CG}}| \times |V_{\text{TG}}|}$ that utilizes the relative spatial distributions of nuclei with respect to tissue regions. For the i^{th} nucleus and j^{th} tissue region, the corresponding assignment is given as,

$$\begin{aligned} &A_{\text{CG}\to\text{TG}}[i,j] = 1, \text{ if } i^{\text{th}} \text{ nucleus centroid} \in j^{\text{th}} \text{ tissue region} \\ &A_{\text{CG}\to\text{TG}}[i,j] = 0, \text{ otherwise} \end{aligned} \tag{5.1}$$

Cell-to-tissue hierarchies for a tissue region are presented in Figure 5.2. Each nucleus is assigned to one and only one tissue region. If a segmented nucleus is at the border of multiple tissue regions, the nucleus is assigned to the tissue region that it has the maximum overlap with. Formally for a given TRoI, a HACT representation is formulated as $G_{\text{HACT}} := \{G_{\text{CG}}, G_{\text{TG}}, A_{\text{CG}\to\text{TG}}\}$.

5.3.3 Graph learning

The HACT graph for a TRoI is processed by a hierarchical GNN to map a TRoI to the corresponding TRoI subtype. To this end, we propose the HierArchical Cell-to-Tissue Network (HACT-Net), a hierarchical GNN architecture described in Figure 5.3.

HACT-Net architecture & learning

HACT-Net intakes G_{HACT} as input and outputs a graph-level representation $h_{\text{HACT}} \in \mathbb{R}^{d_{\text{HACT}}}$. Subsequently, a MLP categorizes h_{HACT}, *e.g.*, to a cancer subtype. Formally, HACT-Net consists of two GNNs, *i.e.*, Cell-GNN (CG-GNN) and Tissue-GNN (TG-GNN), to hierarchically process the HACT graph from fine to coarse level. In this work, we leverage the advances in GNNs and model HACT-Net using PNA layers (Corso et al. (2020)). A thorough description of the PNA architecture is presented in Chapter 2.

First, CG-GNN intakes $G_{\text{CG}} := \{V_{\text{CG}}, E_{\text{CG}}, H_{\text{CG}}\}$, and applies T_{CG} PNA layers to build contextualized cell-node embeddings $h_{\text{CG}}^{(t)}(v)$, $\forall v \in V_{\text{CG}}$. After T_{CG} PNA layers, an LSTM-based jumping knowledge technique (Xu et al., 2018) is employed to adapt to different CG sub-graph structures, *i.e.*,

$$h_{\text{CG}}^{(T_{\text{CG}}+1)}(v) = \text{LSTM}\left(\left\{ h_{\text{CG}}^{(t)}(v) \;\middle|\; t = 1, \ldots, T_{\text{CG}} \right\}\right) \tag{5.2}$$

Following the CG-GNN, the cell-node embeddings, $h_{\text{CG}}^{T_{\text{CG}}+1}(v) \mid v \in V_{\text{CG}}$, and the assignment matrix $A_{\text{CG}\to\text{TG}}$ are used to incorporate hierarchical information and initialize the tissue-node features in the TG, *i.e.*,

$$h_{\text{TG}}^{(0)}(w) = \text{CONCAT}\left(H_{\text{TG}}(w), \sum_{v \in \mathcal{M}(w)} h_{\text{CG}}^{(T_{\text{CG}}+1)}(v)\right) \tag{5.3}$$

where CONCAT denotes concatenation and $\mathcal{M}(w) := \{ v \in V_{\text{CG}} \mid A_{\text{CG}\to\text{TG}}(v, w) = 1 \}$ is the set of nodes in G_{CG} mapping to a node $w \in V_{\text{TG}}$. Analogous to CG, G_{TG} is processed by TG-GNN to compute tissue-node embeddings $h_{\text{TG}}^{(t)}(w)$, $\forall w \in V_{\text{TG}}$. At $t = T_{\text{TG}}$, the embedding of each tissue-node w encodes the cell and tissue information up to T_{TG}-hops from w.

Similar to CG, the tissue-node embeddings in TG are processed via an LSTM-based jumping knowledge technique to combine the intermediate tissue-node embeddings. Finally, the graph-level embedding h_{HACT} is produced by summing all the tissue-node embeddings. A MLP and a softmax operation follows to map h_{HACT} to respective TRoI label. HACT-Net is trained end-to-end by minimizing the cross-entropy loss between the softmax output and the ground-truth TRoI label.

Following Dwivedi et al. (2020), after each PNA layer we include graph normalization (GraphNorm) followed by a batch normalization (BatchNorm). Graph normalization scales the node features by the number of nodes in the graph. Intuitively, it prevents the node representa-

	Metric	N	B	UDH	ADH	FEA	DCIS	I	Total
Image	Number of images	512	758	471	568	783	749	550	4391
	Number of pixels (in million)	2.8	5.7	2.4	2.2	1.2	5.0	8.02	3.9
		±2.7	±4.5	±2.9	±2.0	±1.1	±5.0	±5.4	±4.3
	Max/Min pixel ratio	75.3	97.9	180.1	75.3	58.3	128.6	62.4	235.6
CG	Number of nodes	994	1826	903	863	470	1723	3609	1468
		±732	±1547	±910	±730	±352	±1598	±2393	±1642
	Number of edges	3759	6103	3371	3098	1738	5728	12490	5102
		±2643	±5420	±3675	±2781	±1395	±5811	±10011	±6089
	Max/Min node ratio	71.9	126.6	133.3	104.2	45.2	161.3	113.6	256.4
TG	Number of nodes	107	217	88	100	45	225	423	172
		±106	±233	±93	±91	±32	±217	±317	±217
	Number of edges	509	1012	393	480	194	1111	2025	815
		±545	±1236	±450	±474	±159	±1123	±1741	±1125
	Max/Min node ratio	169.5	312.5	125.0	178.6	416.7	312.5	101.0	434.8
Image split	Train	342	586	303	405	599	562	366	3163
	Validation	86	87	88	77	85	97	82	602
	Test	84	85	80	86	99	90	102	626
WSI split	Train	67	86	59	38	37	33	41	198
	Validation	28	24	24	28	17	21	19	68
	Test	15	16	20	17	12	16	16	59

Table 5.1 – Key statistics of the BRACS dataset.

tions from being at different scales, for graphs of different sizes. This normalization helps the network to learn discriminative topological patterns when the number of nodes varies significantly within a class.

5.4 Datasets

BRACS dataset

As part of this work, we introduce a new dataset termed as BReAst Cancer Subtyping (BRACS). It contains 4391 TRoIs from 325 H&E breast carcinoma WSIs. The WSIs were selected from the archives of the Department of Pathology at National Cancer Institute- IRCCS-Fondazione Pascale, Naples, Italy. They are scanned with an Aperio AT2 scanner at 0.25 μm/pixel resolution. The TRoIs were selected and annotated using QuPath (Bankhead et al., 2017) as:

- Normal tissue (N): they include two types of epithelial cells, namely luminal and basal myoepithelial cells, and two types of stromal cells, namely interlobular and intralobular stroma.
- Benign tissue (B): they include non-proliferative lesions and proliferative lesions with the exception of UDH, FEA and ADH, which are considered as independent subtypes

(see below). Specifically, benign samples include cyst, apocrine metaplasia, ductal ectasia, squamous metaplasia, atrophy, stromal fibrosis, mastitis, sclerosing adenosis, papilloma, radial scar, and simple and complex fibroadenoma.

- Usual Ductal Hyperplasia (UDH): UDH is characterized by a cohesive proliferation of disorderly distributed but oriented cells. It can have different architectural aspects, *e.g.*, solid pattern, fenestrated pattern and micropapillary pattern. UDH has architectural similarities to ADH and DCIS, although it is not an atypical pattern.

- Flat Epithelial Atypia (FEA): FEA is a proliferative lesion characterized by low grade cytological atypia, cell monomorphism, loss of polarity and orientation with respect to the basement membrane, presence of apical snout, endoluminal secretion and frequent calcification.

- Atypical Ductal Hyperplasia (ADH): ADH is a proliferation of monomorphic cells, which partially fill the ductal space. The possible architectural patterns are solid, cribriform and papillary. Cytologic atypia are similar to those of low-grade DCIS (see below), but the lesion does not extend beyond 2mm or has insufficient architectural atypia involving only partial ducts and lobules.

- Ductal Carcinonma in Situ (DCIS): DCIS is a malignant proliferation of epithelial cells that fills the entire duct, without evidence of stroma invasion. Typically it involves multiple adjacent ductal space. It can have cribriform, solid, papillary and micropapillary patterns.

- Invasive Carcinoma (I): they are characterized by the invasion of tumor cells infiltrating the breast stroma with loss of peripheral myoepithelial cells.

Figure 5.4 presents sample TRoIs from all cancer subtypes in BRACS. Each TRoI was first annotated independently by three pathologists. TRoIs with disagreement were further discussed and annotated by the consensus. Note that the pathologists used the entire WSI context during annotation. Figure 5.5 presents some DCIS samples in BRACS dataset, and highlight the included appearance variability. Such TRoI variability is typical in practice, and were included in BRACS to mimic the real world diagnosis. It ensures a realistic and representative evaluation set, with results readily applicable in the field.

Table 5.1 presents category-wise statistics of the TRoIs in BRACS. The statistics demonstrate a high variation in TRoI dimensions. We also include the statistics for the CG and TG representations constructed by our framework, which indicate a large variation in the size of the entity-graph representations. For evaluations on BRACS, we partition the TRoIs into train, validation, and test sets at the WSI-level, such that two TRoIs from the same WSI do not fall in different sets. The WSI-level splitting was performed randomly, ensuring a comparable number of TRoIs per cancer subtype. Such partitioning aimed for a fair evaluation of the compared methods.

BACH dataset

We evaluated the proposed methodology also on the publicly available Grand Challenge on BreAst Cancer Histology images BACH (Aresta et al., 2019). It consists of 400 training and 100 test images from four breast cancer subtypes, *i.e.*, Normal, Benign, DCIS, and Invasive. All images are acquired using a Leica DM 2000 LED microscope and a Leica ICC50 HD camera. These images are in RGB TIFF format and have a fixed size of 2048×1536 pixels and a pixel scale of 0.42×0.42 μm. Notably, the proposed BRACS presents three major advantages over BACH:

- **Number of images**: The train and test sets of BRACS are nearly 10 times and 6 times the size of the train and test sets of BACH, respectively. The large test set ensures a robust evaluation of the methods.
- **Diverse subtypes**: BRACS includes diagnostically complex precancerous atypical categories, namely ADH and FEA, which represent a major diagnostic challenge because of their high risk of progression to cancer. The seven cancer subtypes in BRACS represent a broad spectrum of breast cancer in histopathology.
- **Large variability**: The aforementioned high variability in BRACS in terms of TRoI appearances and dimensions is clinically more representative, and corresponds to a more realistic scenario of breast cancer subtyping.

5.5 Results

In this section, we comparatively assess the proposed method for breast cancer subtyping. First, we introduce state-of-the-art CNN and GNN baselines, and their implementation schemes. Second, we conduct ablations on BRACS to examine the impact of various components in our framework. Third, we evaluate the classification performance of our method and compare with the baselines, on BRACS and BACH datasets for different classification settings. Finally, we include a comparison of HACT-Net with three independent expert-pathologists.

5.5.1 CNN and GNN baselines for comparative evaluation

• **Single-scale CNN** processes TRoIs at a single magnification. A CNN is trained to predict patch-wise cancer subtypes, and we aggregate the patch-wise predictions to produce a TRoI-level prediction. We experiment with images at three magnifications, *i.e.*, 10×, 20×, and 40×, denoted herein as CNN(10×), CNN(20×), and CNN(40×), using the same network architecture and training scheme. For each scale, we extract patches of size 128×128 pixels with a stride of 64 pixels. The CNN follows the single-scale training procedure by Sirinukunwattana (2018), and patch-wise predictions are aggregated using the Agg-Penultimate strategy by Mercan et al. (2019a). We use transfer learning with a ResNet-50 architecture, pre-trained on ImageNet, as the CNN backbone. Following feature extraction by ResNet-50, a two-layer MLP with 128

channels classifies the patches. To improve the classification, the ResNet-50 parameters are fine-tuned. Adam optimizer (Kingma and Ba (2015)) with 10^{-3} learning rate, a batch size of 16, and a dropout of 0.2 is used to optimize the categorical cross-entropy objective.

• **Multi-scale CNN** processes the TRoIs at multiple scales. We extract concentric patches of size 128×128 pixels from multiple magnifications and follow the "Late fusion with single-stream + LSTM" training procedure from Sirinukunwattana (2018). We operate at two settings, *i.e.*, (10×+20×) and (10×+20×+40×), and denote by prepending Multi-scale CNN in front of each. The patch-wise predictions are aggregated using the Agg-Penultimate strategy by Mercan et al. (2019a). On the concatenated features from the LSTM, we use a two-layer MLP of 128 channels to classify the patches. The training strategy and hyperparameters are the same as Single-scale CNN.

• **CGC-Net** denotes the Cell Graph Convolutional Network (CGC-Net) proposed by Zhou et al. (2019a), and it is the state-of-the-art in classifying CG representations for TRoIs. We construct the CG topology for a TRoI using thresholded kNN strategy presented in Section 5.3.2. We initialize the CG nodes with hand-crafted features, employ the Adaptive GraphSage-based CGC-Net architecture, and follow the training strategy proposed by Zhou et al. (2019a).

• **Patch-GNN** implements the method proposed by Aygunes et al. (2020), which is the state-of-the-art GNN method for classifying patch-graph representations of TRoIs. It incorporates local inter-patch context through a GNN to construct a graph-level features, which is then processed by an MLP to classify the TRoIs. We experiment with Patch-GNN at three scales, *i.e.*, 10×, 20×, and 40×, denoted herein as Patch-GNN (10×), Patch-GNN (20×), and Patch-GNN(40×). At each magnification, we extract patches of size 128×128 to construct a TRoI-specific patch-graph. We employ the network architecture and training strategy proposed by Aygunes et al. (2020).

• **CG-GNN** is provided as a standalone CG-based learning baseline, to compare with our proposed hierarchical learning. CG-GNN uses PNA layers, an LSTM-based jumping knowledge, sum readout, and a two-layer MLP classifier. We follow the CG representation strategy as described in Section 5.3.2.

• **TG-GNN** is provided as a standalone TG-based learning baseline, to compare with our proposed hierarchical learning. TG-GNN uses the same architecture as the CG-GNN, with the node features directly initialized by H_{TG} instead of Equation (5.3).

• **CONCAT-GNN** is provided to evaluate the impact of hierarchical graph representation and learning. CONCAT-GNN utilizes standalone CG and TG representations, respectively, as input to standalone CG-GNN and TG-GNN to produce h_{CG} and h_{TG} graph-level embeddings. The TRoI level embedding is constructed by concatenating the graph-level embeddings, *i.e.*, $h_{\text{CONCAT}} = \text{CONCAT}(h_{\text{CG}}, h_{\text{TG}})$. Finally, a two-layer MLP classifies h_{CONCAT} into a cancer subtype.

5.5.2 Implementation

Graph representations: CG representations use, i) patches of size 72×72, and ii) a CNN of ResNet-34 or ResNet-50 to initialize the node features. TG representations (Section 5.3.2) use, i) patches of size 144×144, and ii) a CNN of ResNet-34 or ResNet-50 to initialize the node features.

Graph architecture and learning: Hyper-parameter search was run to find the optimal CG-GNN, TG-GNN, CONCAT-GNN, and HACT-Net parameters:

- # PNA layers in GNN: [3, 4, 5]
- # MLP layers in a PNA layer: 2
- # channels in a PNA-layer MLP: 64
- Graph-level embedding dimension: 128
- # MLP layers in output classifier: 2
- # channels in output MLP classifier: 128
- Training parameters: Adam optimizer (Kingma and Ba, 2015) with a learning rate of 10^{-3}, batch size of 16, and a categorical cross-entropy objective.

Evaluation metrics: Considering the imbalanced number of TRoIs per class in train, validation, and test sets (see Table 5.1), we evaluate the classification performance using weighted F1-score, an average weighted by the number of true instances for each class. The best weighted F1-scores on the validation set is used as the model selection criteria during the training of each method. To present any sensitivity to initialization, we report the mean and standard deviation of each model on the test set by training them three times using random weight initialization. Further, we present precision, recall, and confusion matrices to indicate the distribution of class predictions.

Computational resources: All the experiments were conducted using PyTorch (Paszke et al., 2019) and the Deep Graph Library (DGL) (Wang et al., 2019a), on NVIDIA Tesla P100 Graphics Processing Unit (GPUs) and POWER9 processors.

5.5.3 Ablation studies

We conduct ablation to evaluate the impact of three major components of our methodology on TRoI classification performance, *i.e.*, (i) node feature initialization, (ii) GNN layer type, and (iii) jumping knowledge technique. Each component is analyzed individually, while fixing the other ones. Ablations are performed on BRACS for classifying the TRoIs into 7-classes.

Impact of node feature initialization

The performance of GNNs eminently rely on the initial node features (Kipf and Welling, 2017). In our context, we analyze the impact of initial morphological features of the nodes with the following three feature initialization schemes:

• **No morphological features:** The nodes of an entity-graph are initialized with only the spatial features. Experiments with this setting demonstrate the impact of standalone graph topology on the classification performance.

• **Hand-crafted morphological features:** The entity-graph nodes are initialized with hand-crafted morphological features as suggested by Zhou et al. (2019a), *i.e.*, (i) *texture features*: difference of average foreground to background; standard deviation, skewness, and mean entropy of intensities; dissimilarity, homogeneity, energy, and angular second moment from Gray-Level Co-occurrence Matrix; and (ii) *shape features*: eccentricity, area, maximum and minimum axis lengths, perimeter, solidity, and orientation. Note that, the hand-crafted features for CG and TG are computed, respectively, from the segmented instances of nuclei and tissue regions.

• **CNN morphological features:** The morphological features of the entity-graph nodes are initialized with CNN features (ResNet-34 pre-trained on ImageNet) extracted from patches around the centroids of the nuclei and tissue regions.

Experimental results in Table 5.2 indicate that the standalone CG topology is more discriminative for cancer subtyping than TG topology. The combination of CG and TG topologies further improves discriminative ability. The best performance achieved with the HACT topology confirms the strength of hierarchical representations. Further, including morphological features significantly improves the classification. The superiority of graphs with CNN-based morphological features indicate the richness of morphological information acquired by CNNs, compared to hand-crafted measures.

Impact of GNN layer type

We investigate the impact of two state-of-the-art GNN layers, *i.e.*, GIN and PNA, on the classification performance. The experiments use CNN-based node feature initialization and LSTM-based jumping knowledge. Results in Table 5.3 demonstrate that GNNs with PNA layers outperform GNNs with GIN layers, for all the four GNN constructions. This can be explained by the higher expressive power of the PNA layer, which is designed to operate on graphs with continuous node features.

	Weighed F1
CG-GNN: No morphological features	45.24±1.51
CG-GNN: Hand-crafted morphological features	48.34±5.22
CG-GNN: CNN morphological features	**55.94±1.01**
TG-GNN: No morphological features	36.81±0.71
TG-GNN: Hand-crafted morphological features	51.62±2.11
TG-GNN: CNN morphological features	**56.62±1.35**
CONCAT-GNN: No morphological features	47.62±1.56
CONCAT-GNN: Hand-crafted morphological features	51.55±1.32
CONCAT-GNN: CNN morphological features	**57.01±2.27**
HACT-Net: No morphological features	48.70±0.16
HACT-Net: Hand-crafted morphological features	52.46±0.19
HACT-Net: CNN morphological features	**61.53±0.87**

Table 5.2 – Ablation: Impact of node features. Mean and standard deviation of 7-class weighted F1-scores. Results expressed in %.

	Weighed F1
CG-GNN: GIN	55.70±0.51
CG-GNN: PNA	**55.94±1.01**
TG-GNN: GIN	55.33±1.36
TG-GNN: PNA	**56.62±1.35**
CONCAT-GNN: GIN	56.20±2.12
CONCAT-GNN: PNA	**57.01±2.27**
HACT-Net: GIN	59.73±1.20
HACT-Net: PNA	**61.53±0.87**

Table 5.3 – Ablation: Impact of GNN layer. Mean and standard deviation of 7-class weighted F1-scores. Results expressed in %.

Impact of jumping knowledge technique

To investigate the impact of jumping knowledge, we experiment with three settings: no jumping knowledge, CONCAT-based, and LSTM-based. LSTM-based technique follows Equation (5.2). Based on this, CONCAT-based technique replaces the LSTM operation with the concatenation operation. The experiments use CNN-based node feature initialization and PNA layers. Results in Table 5.4 demonstrate a generally positive impact of the jumping knowledge technique. Compared to CONCAT, the LSTM-based technique learns better dependencies between GNN layers, thus generates better graph embeddings.

Ablation summary

The ablation experiments conclude the following choice of components for designing our methodology, i) CNN-based initialization of node-level morphological features, ii) use of PNA

	Weighed F1
CG-GNN: No aggregator	55.53±0.75
CG-GNN: Concatenation	55.82±0.97
CG-GNN: LSTM	**55.94±1.01**
TG-GNN: No aggregator	55.30±0.81
TG-GNN: Concatenation	56.07±0.80
TG-GNN: LSTM	**56.62±1.35**
CONCAT-GNN: No aggregator	**57.67±4.66**
CONCAT-GNN: Concatenation	56.28±2.75
CONCAT-GNN: LSTM	57.01±2.27
HACT-Net: No aggregator	49.16±1.15
HACT-Net: Concatenation	59.78±1.59
HACT-Net: LSTM	**61.53±0.87**

Table 5.4 – Ablation: Impact of GNN jumping knowledge technique. Mean and standard deviation of 7-class weighted F1-scores. Results expressed in %.

layers, and iii) an LSTM-based jumping knowledge technique.

5.5.4 Classification results on BRACS dataset

We evaluate our proposed methods, comparatively with CNN and GNN baselines. To analyze the performance for different clinical applications and histopathological needs, we evaluate and report the results separately in the following three settings:

• **Setting 1: 7-class classification:** Here, we classify the TRoIs into 7-classes, *i.e.*, Normal, Benign, UDH, ADH, FEA, DCIS, and Invasive, for the differentiation of a large spectrum of breast cancer subtypes. Table 5.5 tabulates the classification performance of the compared methods.

Among single-scale CNNs, CNN(10×) performs the best, indicating the importance of global context information for TRoI classification. Multi-scale CNNs using both global and local context outperform single-scale CNNs. Such benefit from context is significant for ADH, FEA, and DCIS categories, which all require both local and global context for the diagnosis. Multi-scale CNNs also outperform CGC-Net and Patch-GNNs. Interestingly, at each magnification, Patch-GNN outperforms single-scale CNN, which affirms the importance of relational and topological information incorporated in the graphs.

Comparing our proposed GNN solutions, we observe that CG-GNN significantly outperforms CGC-Net, indicating the superiority of CNN-based node feature initialization over handcrafted features, and the significance of GNNs with expressive PNA layers over Adaptive GraphSage in CGC-Net. We notice that CG-GNN and TG-GNN provide comparable performance overall. However, they outperform each other for Normal, Benign, UDH, ADH, and FEA categories, displaying the importance of complementary information captured by standalone TG and

	Method	Normal	Benign	UDH	ADH	FEA	DCIS	Invasive	WF1
CNN	CNN (10×)	48.67	44.33	45.00	24.00	47.00	53.33	86.67±2.64	50.85
		±1.71	±1.89	±4.97	±2.83	±4.32	±2.62	±2.64	±2.64
	CNN (20×)	42.00	42.33	39.33	22.67	47.67	50.33	77.00	46.85
		±2.16	±3.09	±2.05	±2.49	±1.25	±3.09	±1.41	±2.19
	CNN (40×)	32.33	39.00	23.67	18.00	37.67	47.33	70.67	39.41
		±4.64	±0.82	±1.70	±0.82	±2.87	±2.05	±0.47	±1.89
	Multi-scale CNN (10 + 20×)	48.33	45.67	41.67	32.33	46.33	59.33	85.67	52.27
		±2.05	±0.47	±4.99	±0.94	±1.41	±2.05	±1.89	±1.93
	Multi-scale CNN (10 + 20 + 40×)	50.33	44.33	41.33	31.67	51.67	57.33	86.00	52.83
		±0.94	±1.25	±2.49	±3.30	±3.09	±0.94	±1.41	±1.92
GNN	CGG-Net	30.83	31.63	17.33	24.50	58.97	49.36	75.30	43.63
		±5.33	±4.66	±3.38	±5.24	±3.56	±3.41	±3.20	±0.51
	Patch-GNN (10×)	52.53	47.57	23.67	30.66	60.73	58.76	81.63	52.10
		±3.27	±2.25	±4.65	±1.79	±5.35	±1.15	±2.17	±0.61
	Patch-GNN (20×)	43.86	43.37	19.47	25.73	55.57	52.86	79.20	47.10
		±4.23	±3.21	±2.31	±2.87	±2.08	±1.85	±1.04	±0.70
	Patch-GNN (40×)	41.70	32.93	25.07	25.63	49.47	48.60	71.57	43.23
		±3.06	±1.04	±3.74	±2.01	±3.46	±4.23	±5.15	±0.57
Ours	CG-GNN	58.77	40.87	**46.82**	39.99	63.75	53.81	81.06	55.94
		±6.82	±3.05	±**1.95**	±3.56	±10.48	±3.89	±3.33	±1.01
	TG-GNN	**63.59**	**47.73**	39.41	28.51	72.15	54.57	82.21	56.62
		±**4.88**	±**2.87**	±4.70	±4.29	±1.35	±2.23	±3.99	±1.35
	CONCAT-GNN	60.97	43.06	41.96	26.10	71.29	60.83	85.42	57.01
		±4.54	±2.26	±4.67	±3.73	±2.09	±3.71	±2.70	±2.27
	HACT-Net (Proposed)	61.56	47.49	43.60	**40.42**	**74.22**	**66.44**	**88.40**	**61.53**
		±2.15	±2.94	±1.86	±**2.55**	±**1.41**	±**2.57**	±**0.19**	±**0.87**

Table 5.5 – Mean and standard deviation of per-class F1-scores and weighted F1-scores (WF1) for 7-class classification setting. Results are expressed in %. The best result is in **bold** and the second best is underlined.

CG representations. Further, both HACT-Net and CONCAT-GNN provide overall superior performance compared to all CNN and GNN baselines. HACT-Net significantly outperforms CONCAT-GNN showing the significance of hierarchical modeling and learning. CONCAT-GNN produces overall comparable or superior performance to CG-GNN and TG-GNN, although for individual classes, CONCAT-GNN is rarely better than the two, suggesting that it may be using complementary information from CG and TG. Such complementary information is indeed best utilized by HACT-Net, with high per-class and overall classification performance. Though HACT-Net achieves the third best result for the UDH category, it uses the complementarity of CG and TG to provide better classification than TG-GNN. Moreover, the misclassified UDH samples are predominantly categorized as Benign due to the expected ambiguity between Benign and UDH classes. All the proposed GNNs often outperform all CNN baselines, establishing the potential of entity-based analysis.

Figure 5.6 and Figure 5.7 present per-class precision and recall for CG-GNN, TG-GNN, CONCAT-

	Method	Normal	Non-cancerous	Precancerous	Cancerous	Weighted F1
CNN	CNN (10×)	54.33±3.68	56.00±0.82	56.33±1.25	83.67±0.94	64.36±1.37
	CNN (20×)	45.33±4.64	55.33±0.47	52.33±1.89	81.67±2.05	61.18±1.93
	CNN (40×)	42.00±4.89	51.00±0.82	47.67±4.11	77.67±2.05	56.99±2.72
	Multi-scale CNN (10 × +20×)	51.67±5.79	55.33±1.25	52.67±2.87	80.67±1.89	61.82±2.53
	Multi-scale CNN (10 × +20 × +40×)	51.33±3.27	56.33±2.05	57.00±1.64	81.33±3.68	63.52±2.59
GNN	CGG-Net	34.53±2.93	47.23±3.72	62.90±2.81	82.20±1.04	59.87±2.30
	Patch-GNN (10×)	53.13±4.40	46.23±2.45	63.96±3.82	77.43±3.22	61.93±2.51
	Patch-GNN (20×)	53.46±1.81	47.16±2.81	63.20±3.78	74.90±3.36	61.26±2.90
	Patch-GNN (40×)	40.90±2.75	38.67±2.76	56.77±3.91	72.20±2.61	54.60±1.90
Ours	CG-GNN	52.95±12.11	<u>56.55±3.70</u>	61.53±3.03	<u>84.47±0.87</u>	66.10±2.58
	TG-GNN	52.96±6.81	56.52±2.85	<u>64.36±1.05</u>	82.21±0.78	<u>66.24±1.11</u>
	CONCAT-GNN	<u>54.54±1.64</u>	**56.63±1.68**	62.58±1.45	81.80±0.77	65.83±0.04
	HACT-Net (Proposed)	**66.08±3.69**	55.28±1.74	**66.21±0.87**	**84.91±0.79**	**69.04±0.46**

Table 5.6 – Mean and standard deviation of per-class F1-scores and weighted F1-scores for 4-class classification setting. Results are expressed in %. The best result is in **bold** and the second best is <u>underlined</u>.

GNN, and HACT-Net. HACT-Net produces the highest precision values for most of the classes. The recall ranking between CG-GNN and TG-GNN varies across classes, whereas HACT-Net consistently yields good recall values. Further, standard deviation of class-wise precision and recall values are the lowest for HACT-Net, for most classes. Figure 5.8 presets row-normalized aggregated 7-class confusion matrix across three runs for HACT-Net. It indicates ambiguities between (i) Normal and Benign, (ii) UDH and ADH, and (iii) ADH and DCIS. Notably, these pair-wise classes bear high pathological ambiguity and are diagnostically challenging.

• **Setting 2: 4-class classification:** This setting categorizes TRoIs into 4-classes as per cancer risk: Normal, Non-cancerous (Benign + UDH), Precancerous (ADH + FEA), and Cancerous (DCIS + Invasive). Classification performance of CNN and GNN baselines, and HACT-Net are presented in Table 5.6. Single scale CNNs exhibit the same behavior as in the 7-class setting. However, combining multiple magnifications in multi-scale CNNs does not improve the classification over the single-scales. Among the baselines, CGC-Net and Patch-GNNs perform comparable or inferior to the CNNs, with a low-magnification CNN(10×) outperforming the others. Similarly to the 7-class setting, our proposed methods are superior to the baselines. HACT-Net produces the best overall performance, with the best classification performance for Normal, Precancerous, and Cancerous categories. To highlight, HACT-Net achieves ≈ 66% F1-score for the diagnostically challenging Precancerous category.

• **Setting 3: Binary classifications:** In this setting, we replicate the typical decision process of a pathologist for breast cancer subtyping which follows a diagnostic decision tree as presented in Figure 5.9. It is inspired by the classification scheme presented by Mercan et al. (2018). Note that such individual binary decisions are less constrained compared to multi-class classification, thus allows for better discrimination between a selected pair of classes. The binary

	Method	I vs N+B+A+U+F+D	N+B+U vs A+F+D	N vs B+U	B vs U	A+F vs D	A vs F	Aggregated
CNN	CNN (10×)	95.66	81.24	69.83	76.12	73.44	77.59	78.90
		±0.48	±0.42	±0.38	±1.13	±2.56	±1.73	±1.38
	CNN (20×)	92.39	80.84	66.52	74.75	67.87	71.78	75.69
		±0.37	±0.36	±2.14	±1.51	±1.82	±2.53	±1.68
	CNN (40×)	90.74	79.92	62.36	68.13	64.86	66.91	72.15
		±0.59	±1.66	±2.14	±4.30	±2.98	±1.68	±2.51
	Multi-scale CNN (10 + 20×)	94.31	80.89	67.99	75.58	72.07	76.91	77.96
		±1.26	±1.31	±1.86	±2.06	±1.85	±2.22	±1.80
	Multi-scale CNN (10 + 20 + 40×)	95.12	82.21	70.87	72.89	72.08	75.47	78.11
		±1.15	±0.34	±2.07	±2.26	±3.17	±3.69	±2.40
GNN	CGG-Net	91.60	79.73	63.67	62.37	81.56	73.80	75.46
		±2.09	±1.53	±3.12	±3.00	±1.56	±5.41	±3.09
	Patch-GNN (10×)	95.80	76.53	72.57	72.87	77.17	78.26	78.87
		±0.43	±0.32	±1.10	±3.07	±0.85	±2.60	±1.75
	Patch-GNN (20×)	93.70	76.63	70.10	69.77	74.10	81.03	77.55
		±0.36	±1.40	±1.90	±3.13	±0.10	±1.85	±1.78
	Patch-GNN (40×)	92.40	74.43	71.10	67.40	72.97	76.40	75.78
		±0.95	±0.64	±1.74	±2.46	±0.66	±1.95	±1.56
Ours	CG-GNN (Ours)	94.52	**83.79**	75.71	73.15	77.48	84.33	81.50
		±0.43	**±0.31**	±1.68	±3.32	±1.68	±0.54	±1.70
	TG-GNN	96.00	80.38	69.51	76.12	80.67	84.18	81.14
		±0.80	±3.12	±0.99	±0.22	±3.56	±2.02	±1.34
	CONCAT-GNN	95.91	83.210	71.84	75.67	80.14	88.88	82.61
		±0.56	±0.68	±1.46	±1.81	±2.60	±3.86	±2.15
	HACT-Net (Proposed)	**96.32**	83.63	**76.84**	**77.66**	**81.11**	**89.35**	**84.15**
		±0.64	±0.73	**±0.68**	**±0.37**	**±0.72**	**±0.26**	**±0.60**

Table 5.7 – Mean and standard deviation of weighted F1-scores for binary classification setting. Further, the aggregated mean and standard deviation for the six binary tasks are reported. Results are expressed in %. The best result is in **bold** and the second best is underlined.

classifiers can assist pathologists in categorizing ambiguous cases at different bifurcations of the decision tree. Table 5.7 presents the results for six individual binary classifications, at the bifurcations in the decision tree. Results are consistent with the 7-class and 4-class classification settings, with HACT-Net consistently outperforming all baselines and providing the best aggregated score.

Domain expert comparison on BRACS dataset

To further benchmark our proposed methodology as well as to assess the quality of the introduced BRACS dataset, we acquired annotations of the BRACS test set from additional independent pathologists. For such comparison with domain experts, we follow the evaluation protocol in Elmore et al. (2015). We recruited three board-certified pathologists (other than the three pathologists who provided the initial annotations, namely our ground truth labels), from

	Normal	Benign	UDH	ADH	FEA	DCIS	Invasive	WF1	Acc
Pathologist 1	67.53	53.92	41.90	36.00	19.13	71.59	94.00	55.30	56.71
Pathologist 2	47.83	52.94	25.00	35.37	65.22	68.00	94.00	57.07	57.99
Pathologist 3	39.66	49.59	49.43	42.29	54.12	65.19	89.47	56.71	56.55
Pathologist statistics	51.57	**52.15**	38.78	37.89	46.16	**68.26**	**92.49**	56.36	57.08
	±11.70	±**1.85**	±10.22	±3.12	±19.64	±**2.62**	±**2.14**	±0.76	±0.64
HACT-Net statistics	**61.56**	47.49	**43.60**	**40.42**	**74.22**	66.44	88.40	**61.53**	**63.21**
	±**2.15**	±2.94	±**1.86**	±**2.55**	±**1.41**	±2.57	±0.19	±**0.87**	±**0.27**

Table 5.8 – Comparison between HACT-Net and domain expert pathologists for 7-class breast cancer subtyping on BRACS dataset. Per-class F1-scores, weighted F1-scores (WF1) and accuracy (Acc) for 7-class classification are presented. Results are expressed in %. The best results are in **bold**.

	Pathologist 1	Pathologist 2	Pathologist 3	Ground truth
Pathologist 1	-	47.60	50.96	56.71
Pathologist 2	-	-	64.38	57.99
Pathologist 3	-	-	-	56.55

Table 5.9 – Concordance among three independent pathologists for annotating BRACS test dataset. Results are expressed in %.

three different medical centers, to further ensure independence. Namely, National Cancer Institute- IRCCS-Fondazione Pascale, Naples, Italy; Lausanne University Hospital, CHUV, Lausanne, Switzerland; and Aurigen, Centre de Pathologie, Lausanne, Switzerland. These experts are specialized in breast pathology and have been in practice for over twenty years. The pathologists independently and remotely annotated BRACS test set TRoIs, without having access to respective WSIs. This protocol ensures equal field-of-view for all the pathologists as well as our methodology.

The independent pathologists' annotations are compared to the ground truth, with the results shown in Table 5.8. We present per-class F1-scores, overall weighted F1-score, and overall accuracy for each pathologist. We also include the aggregated statistics of the three pathologists for benchmarking HACT-Net with domain experts. Table 5.8 indicates that HACT-Net outperforms the domain experts in distinguishing TRoIs of diagnostically challenging classes, *i.e.*, atypia and hyperplasia, while yielding comparable performance for the normal and cancerous categories. Per-class standard deviations of pathologists' statistics show the expected high inter-observer variability in breast cancer diagnosis. Compared to the pathologists, HACT-Net yields a superior weighted accuracy and weighted F1 given the ground truth diagnoses for the 7-class classification.

To benchmark the BRACS dataset with respect to the dataset by Elmore et al. (2015), we compare the aggregated pathologist statistics on both datasets for the same set of classes, *i.e.*, Benign without atypia (Normal + Benign + UDH), Atypia (ADH + FEA), DCIS, and Invasive. Note that the dataset by Elmore et al. (2015) consists of 240 breast biopsy slides, while BRACS

consists of 626 TRoI images. For the dataset by Elmore et al. (2015), class-wise concordance rates (class-weighted average accuracy of 115 pathologists to a three-expert consensus) are 87%, 48%, 84%, and 96%, respectively for the four aforementioned classes. For BRACS, the similar class-wise concordance rates are 87%, 50%, 72%, and 90%, respectively. The class-wise concordance rates exhibit a similar trend in both datasets. Differences can be attributed to differing fields-of-view, *i.e.*, TRoI vs. WSI, accessible to the pathologist during annotation.

Table 5.9 presents the inter-observer concordance rates for the BRACS test set. We notice significant differences in the concordance rates between pathologists 2 vs.3 and pathologist 1 vs. the other two. This can be reasoned to differing histopathology practices across different regions.

Computational time analysis

We report computational time for processing a tumor RoI of size 1000×1000 pixels on a single-core POWER8 processor combined with an NVIDIA P100 GPU. Stain normalization with the Macenko method takes 0.8 seconds (Central Processing Unit (CPU)-only), CG generation 2.51 seconds, and TG generation 4.14 seconds. Overall, the computational time for transforming the Region-of-Interest (RoI) into HACT representation is 7.92 seconds. The superpixel extraction for constructing the graph representation can be further optimized by using fast GPU implementations, *e.g.*, as proposed by Jampani et al. (2018). Provided the HACT representation, HACT-Net provides near real-time inference by requiring 34.11 milliseconds.

5.5.5 Classification results on BACH dataset

We evaluate the methods on the public BACH dataset. Considering its smaller training set of 400 images, we employ different image augmentation techniques for training HACT-Net. To this end, we employ rotation, mirroring, and color augmentations on the training images before extracting HACT graph representations. We do not use other graph augmentation techniques, such as random node and edge dropping, since these augmentations may hamper the meaningful topological distribution of the biological entities. The implementation strategies and hyperparameters in Section 5.5.2 are employed for training HACT-Net. Classification performance of HACT-Net and the current state-of-the-art results on the BACH dataset are listed in Table 5.10. Our predictions have been evaluated independently by the organizers of the BACH challenge, ensuring a fair comparison. HACT-Net results in comparable classification accuracy with the state-of-the-art methods. The difference in the accuracies are not significant considering only 100 TRoIs in the test set. Notably, our methodology employs a single, unified network, where the other listed competitors employ an ensemble strategy with multiple networks during inference.

	Methods	Accuracy
Ensemble networks	Wang et al. (2019)	95.00
	Marami et al. (2018)	94.00
	Yang et al. (2019)	93.00
	Chennamsetty et al. (2018)	87.00
	Kwok et al. (2018)	87.00
	Brancati et al. (2018)	86.00
Single network	HACT-Net	91.00

Table 5.10 – Accuracy of 4-class breast cancer subtyping in BACH dataset. Results are expressed in %.

5.5.6 Qualitative analysis

Qualitative assessment of a few TRoIs from the BRACS dataset using HACT-Net, CG-GNN, and TG-GNN is presented in Figure 5.10. In Figure 5.11, we use GRAPHGRADCAM (Pope et al., 2019; Jaume et al., 2021b), as presented in Chapter 3, a post-hoc gradient based feature attribution technique, to highlight the nuclei and tissue-region nodes in CG and TG, respectively, which our model focuses on while classifying the TRoIs. Given the DCIS examples in Figure 5.11(a-c&g-i), HACT-Net is seen to focus on the diagnostically relevant tumorous epithelium and necrotic regions in TG, while ignoring the less important stroma and lumen, cf. Figure 5.11(b,h). Further, within the relevant tissue regions, HACT-Net focuses on a subset of tumorous epithelial nuclei in CG, as shown in Figure 5.11(c,i). Interestingly, we observe in Figure 5.11(h,i) that HACT-Net uses complementary information from the necrotic region captured by TG, but not by CG. Similar observations of HACT-Net considering the diagnostically relevant regions can be made for FEA and Benign examples shown in Figure 5.11(d-f&j-l). Noticeably, such feature attribution analysis of GNNs localizes and highlights the focus of deep networks in the given entity-paradigm, which is both more interpretable and more explainable compared to feature attribution strategies in a pixel-paradigm (Jaume et al., 2020, 2021b) (see Chapter 6). Interestingly, we also analyze the impact of tissue or slide preparation artifacts on the model performance. In Figure 5.12, we present a DCIS image with tissue-tear and blur artifacts. We observe that the detected superpixels do not aptly depict the tissue in the blur region, and consequently the TG-GNN using standalone TG misclassifies it. However, the nuclei detection is less impacted by the artifact, which allows the CG to appropriately encode the cell microenvironment and correctly classify the sample. To highlight, HACT-Net utilizing the complementary information from both CG and TG compensates for the issue in TG, and correctly identifies the subtype.

5.6 Conclusion

Pixel-based processing of pathology images suffers from the context-resolution trade-off, and misses the notion of biological entity and tissue composition. In this chapter, we presented an

entity-based tissue representation and learning to address these issues. To that end, our two specific contributions are: (i) a hierarchical entity-graph representation of a tissue image by incorporating multisets of pathologically intuitive biological entities, and (ii) a hierarchical graph neural network for sequentially processing the entity-graph representation for mapping tissue compositions to tissue subtypes. Further, we introduced BReAst Cancer Subtyping (BRACS), a large cohort of breast tumor regions-of-interest, annotated with breast cancer subtypes. BRACS encompasses seven breast cancer subtypes to present a realistic breast cancer diagnosis scenario. Using BRACS as well as a public breast cancer subtyping dataset BACH, we demonstrated herein the superior performance of our proposed methodology for classifying breast tumor regions-of-interest into cancer subtypes. Under various experimental settings, our methodology is shown to outperform state-of-the-art pixel-based and entity-graph based classification approaches. Furthermore, we benchmarked our methodology on the BRACS dataset by comparing it to three independent pathologists. Notably, our method achieved better performance for per-cancer subtype and overall aggregated classification. Although we have evaluated our method for breast cancer classification, the technology is easily extendable to other tissue types and diseases. Notably, the proposed hierarchical graph methodology can also be adapted to other image modalities, such as natural images, multiplexed images, hyperspectral images, satellite images, and other medical imaging domains, by utilizing domain and task-specific entities.

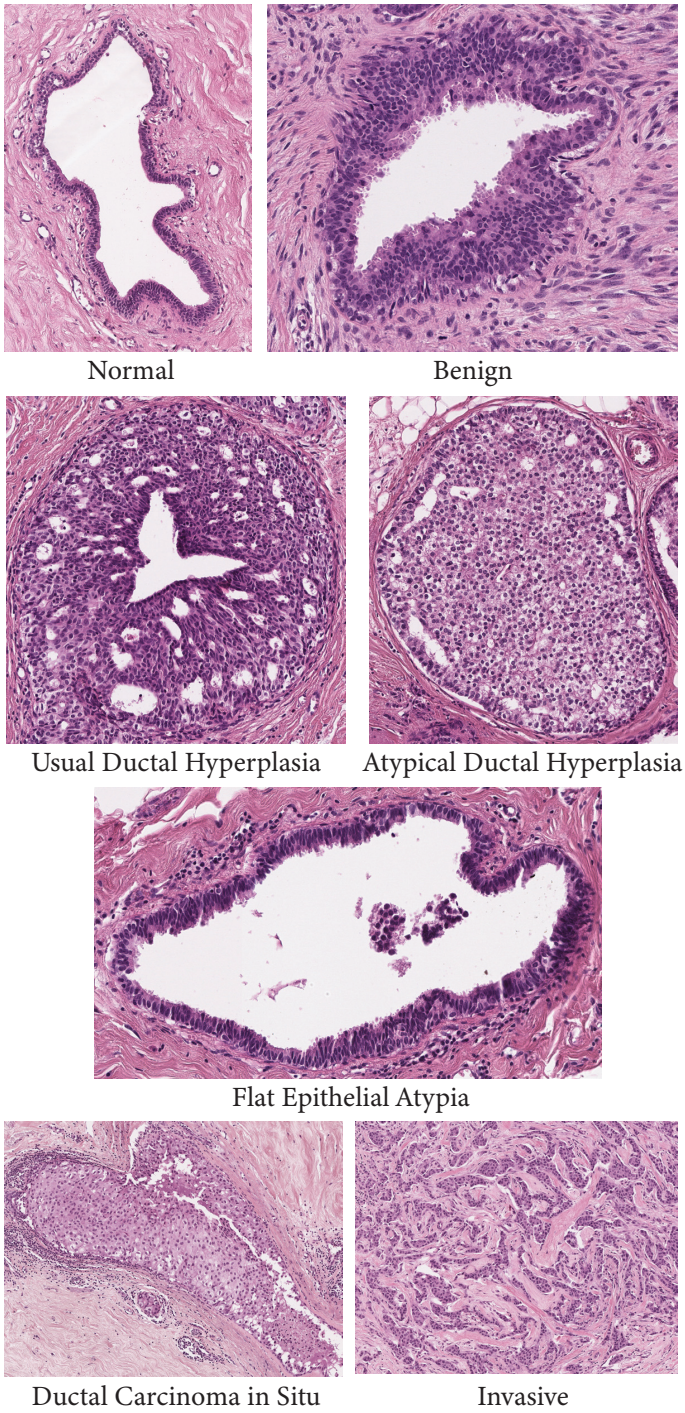


Figure 5.4 – Samples of class-wise TRoI in BRACS dataset. (Figure is best viewed in color.)

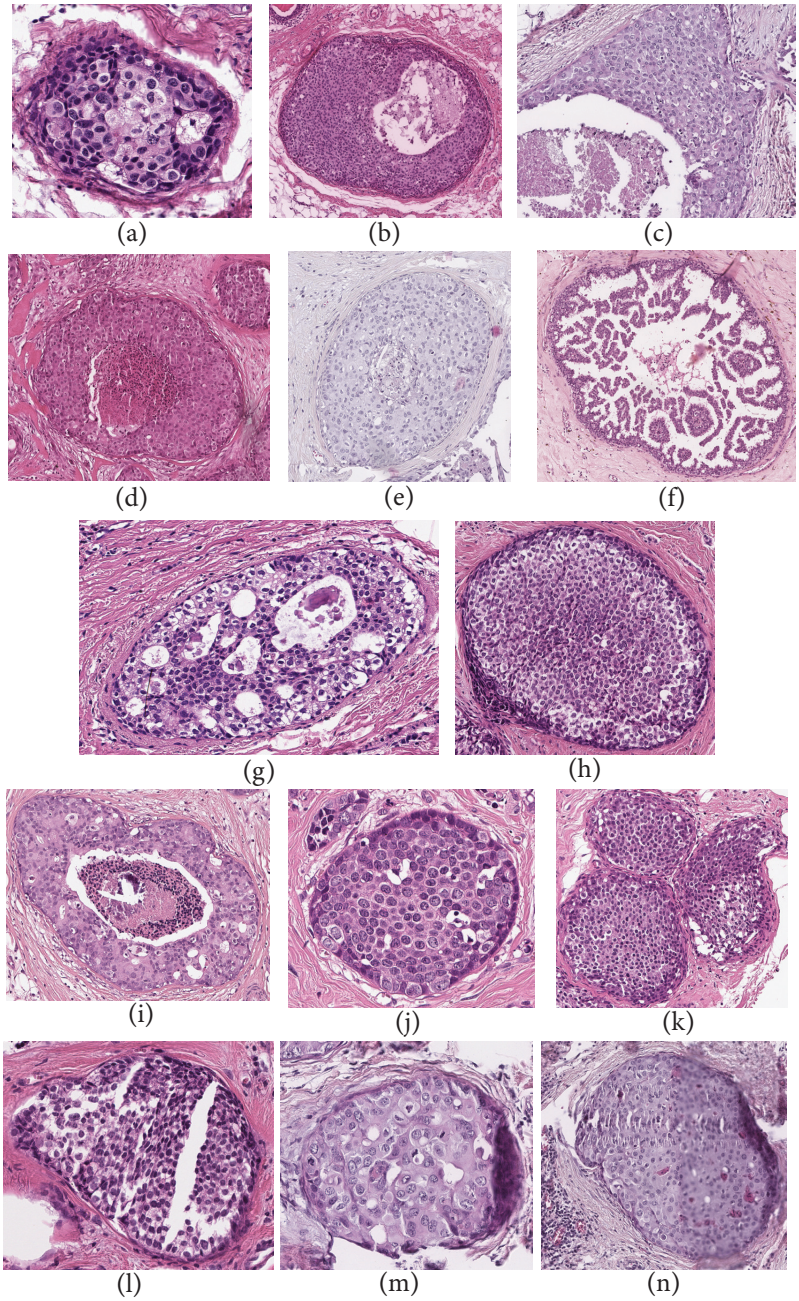


Figure 5.5 – Overview of the variability for DCIS category in BRACS. The samples depict variations in, (a, b, c) tumor size, (d, e) staining appearance, sub-patterns: (f) low-grade papillary, (g) moderate-grade cribriform, (h, i) high-grade solid and comedo, (j, k) number of glandular regions per TRoI, and artifacts due to tissue and slide preparation: (l) tissue-folding or tear, (m) ink stain, (n) blur. Similar variability also persists in other categories in BRACS. (Figure is best viewed in color.)

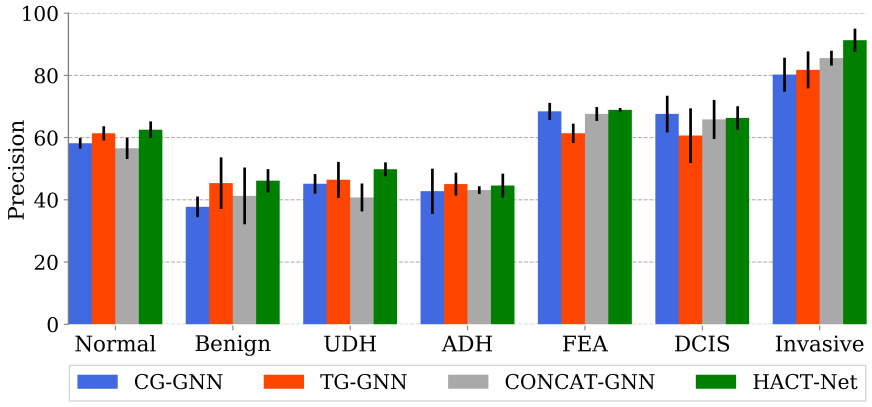


Figure 5.6 – Mean and standard deviation of per-class precision for 7-class classification with HACT-Net. (Figure is best viewed in color.)

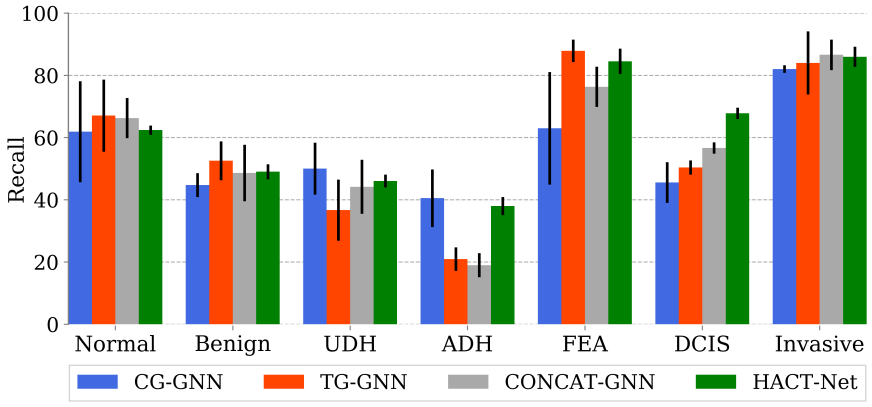


Figure 5.7 – Mean and standard deviation of per-class recall for 7-class classification with HACT-Net. (Figure is best viewed in color.)

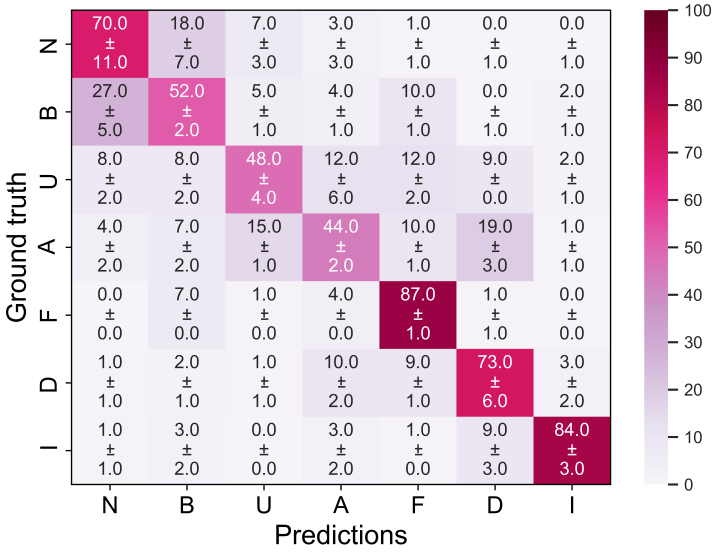


Figure 5.8 – Mean and standard deviation of row-normalized 7-class confusion matrix for HACT-Net.

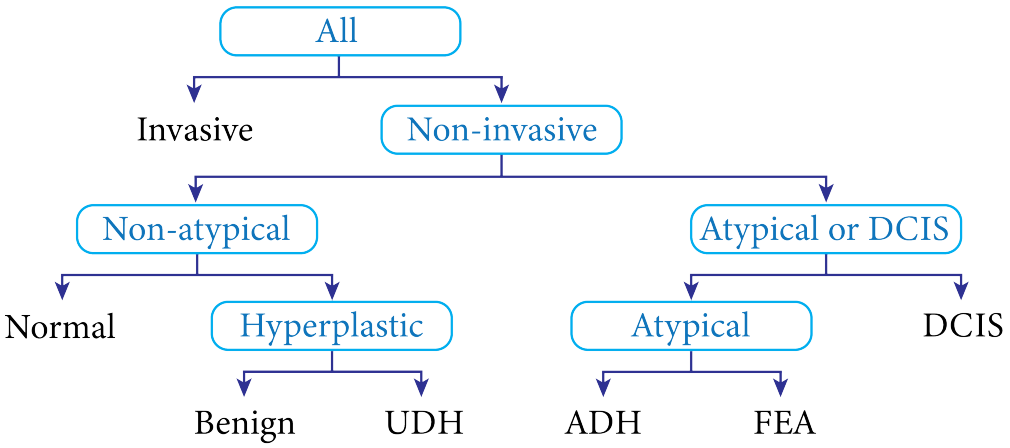


Figure 5.9 – Decision tree used by pathologists for breast cancer diagnosis. The 7-class classification is simplified to a series of binary decision tasks, through which the diagnosis becomes more and more specific until the leaves, *i.e.*, the 7 diagnostic decision classes, are reached.

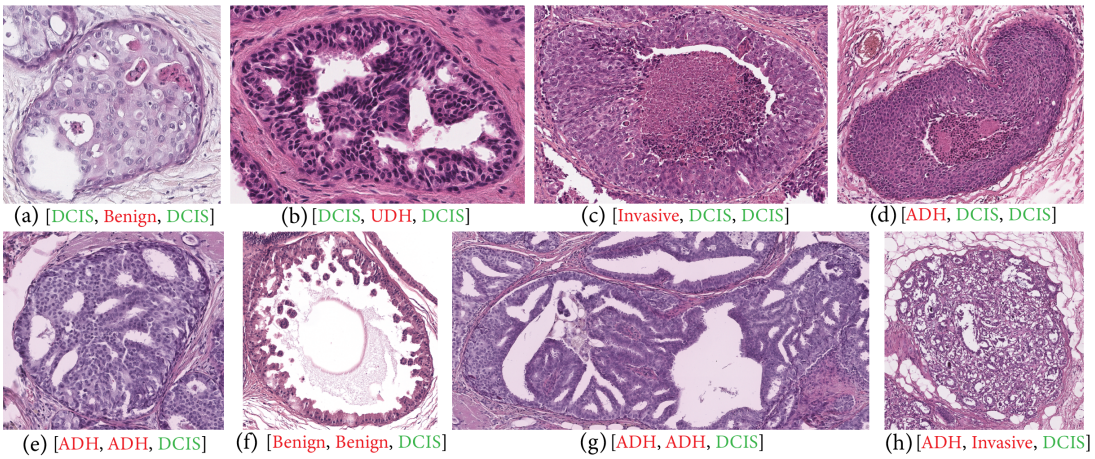


Figure 5.10 – Qualitative comparison of CG-GNN, TG-GNN, and HACT-Net for 7-class classification. Predictions by the classifiers are noted below each example. Red and Green denote incorrect and correct classification, respectively. (a,b) TRoI which TG-GNN misclassifies, while CG-GNN and HACT-Net classify correctly by using the nuclei characteristics. (c,d) TRoI misclassified by CG-GNN, while correctly classified by TG-GNN and HACT-Net by using context information from necrotic regions. (e,f,g,h) TRoI which both CG-GNN and TG-GNN misclassify, where HACT-Net classifies correctly by utilizing both cell and tissue microenvironments together. (Figure is best viewed in color.)

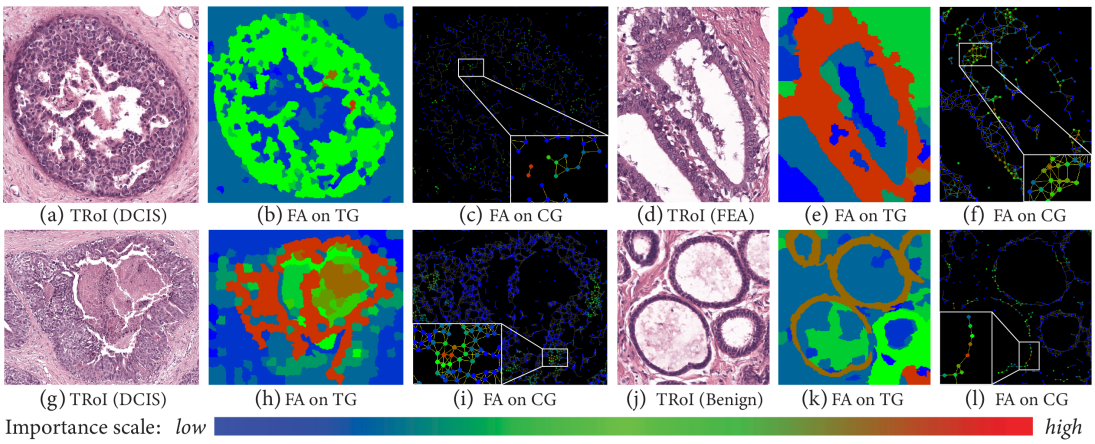


Figure 5.11 – Feature attribution (FA) maps of HACT-Net on TG and CG for four sample TRoIs for 7-class classification: Sample TRoIs of (a,g) DCIS, (d) FEA, and (j) Benign classes, with their corresponding feature attribution maps on (b,h,e,k) TG and (c,i,f,l) CG. (Figure is best viewed in color.)

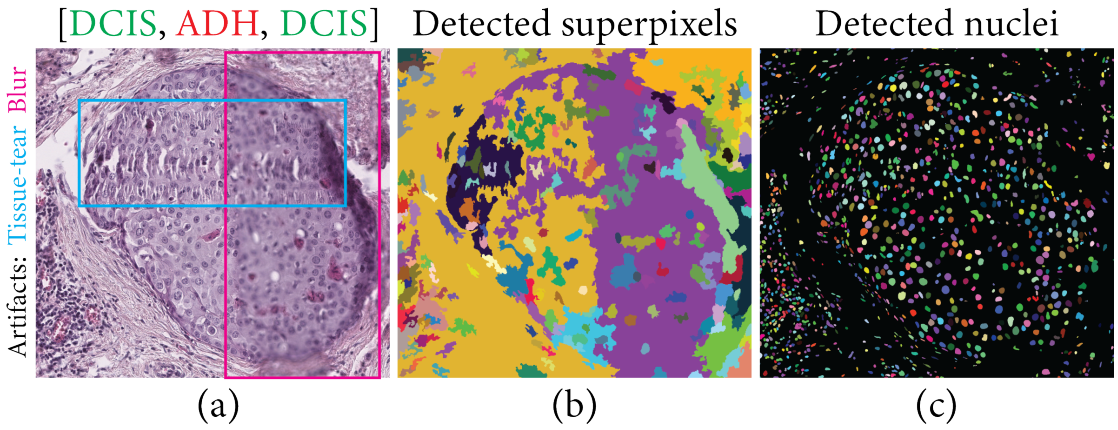


Figure 5.12 – (a) A DCIS sample including tissue-tear and blur artifacts. (b) Detected superpixels. (c) Detected nuclei. The classifications by CG-GNN, TG-GNN and HACT-Net are indicated, where Red and Green denote incorrect and correct classification.

6 Quantifying Explainers of Graph Neural Networks in Computational Pathology

The ideas, methods and results presented in this chapter are published in:

- "Quantifying Explainers of Graph Neural Networks in Computational Pathology", **Guillaume Jaume***, Pushpak Pati*, Behzad Bozorgtabar, Antonio Foncubierta, Anna Maria Anniciello, Florinda Feroce, Tilman Rau, Jean-Philippe Thiran, Maria Gabrani, Orcun Goksel. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021 (Jaume et al., 2021b).
- "Towards Explainable Graph Representations in Digital Pathology", **Guillaume Jaume***, Pushpak Pati*, Antonio Foncubierta, Florinda Feroce, Giosue Scognamiglio, Anna Maria Anniciello, Jean-Philippe Thiran, Orcun Goksel, Maria Gabrani. In *International Conference on Machine Learning (ICML), ICML Workshop on Computational Biology*, 2020 (Jaume et al., 2020).

GJ (the author of this thesis) is sharing first co-authorship with PP. The ideas, concepts and experiments were designed by GJ and PP. AMA, FF and TR shaped the medical aspects of the work, in order to define appropriate nuclei-level attributes, understand the expected nuclei appearance, and to study the agreement between the pathologists and the AI model. JPT and OG supervised and supported GJ in organizing his research. The manuscript was written by GJ and PP and subsequently revised by BB, AF and OG.

6.1 Introduction

Histopathological image understanding has been revolutionized by recent machine learning advancements, especially DL (Bera et al., 2019; Serag et al., 2019). DL has catered to increasing diagnostic throughput as well as a need for high predictive performance, reproducibility and objectivity. However, such advantages come at the cost of a reduced transparency in decision-making processes (Hagele et al., 2020; Holzinger et al., 2017; Tizhoosh and Pantanowitz, 2018). Considering the need for reasoning any clinical decision, it is imperative to enable the

explainability of DL decisions to pathologists.

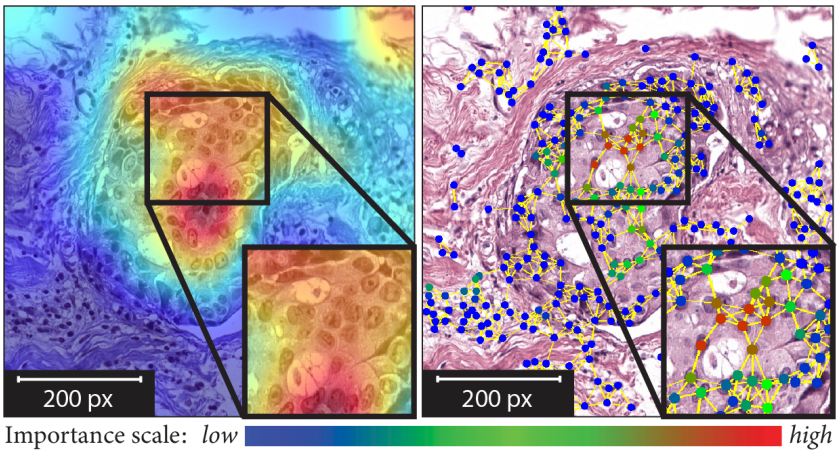


Figure 6.1 – Sample explanations produced by pixel- and entity-based explainability techniques for a ductal carcinoma *in situ* (DCIS) TRoI.

Inspired by the explainability techniques (explainers) for DL model decisions on natural images (Selvaraju et al., 2017; Chattopadhay et al., 2018; Simonyan et al., 2013; Zeiler and Fergus, 2014; Yosinski et al., 2015; Bach et al., 2015; Montavon et al., 2015; Kindermans et al., 2015; Zintgraf et al., 2017; Kim et al., 2018), several explainers have been implemented in digital pathology, such as feature attribution (Hagele et al., 2020; Bruno et al., 2017; Binder et al., 2018), concept attribution (Graziani et al., 2020), and attention-based learning (Lu et al., 2021b). However, pixel-level explanations, as shown in Figure 6.1, pose several notable issues, including: (i) a pixel-wise analysis disregards the notion of biological tissue entity, their topological distribution, and the inter-entity interactions; (ii) a typical patch-based DL processing and explainer fail to integrate complete tumor macro-environment information; and (iii) pixel-wise visual explanations, *i.e.*, heatmaps of salient regions, tend to be blurry and hard to interpret. Explainability in entity space is thus a natural choice to address the aforementioned issues. To that end, we propose to transform the original histology image into an *entity-graph* representation, where nodes and edges denote biological entities and inter-entity interactions, respectively. The choice of entities, such as cells (Gunduz et al., 2004; Zhou et al., 2019a; Pati et al., 2021b), tissues (Pati et al., 2021b) or others, can be task-dependent. Then, we learn a GNN (Kipf and Welling, 2017; Xu et al., 2019b) to model the *entity-graph*. Subsequently, explainers for graph-structured data Pope et al. (2019); Ying et al. (2019); Baldassarre and Azizpour (2019) applied to the entity-graphs highlight responsible entities for the concluded diagnosis, thereby generating intuitive explanations for pathologists.

In the presence of various graph explainers producing distinct explanations for an input, it is crucial to discern the explainer that best fits the explainability definition (Arrieta et al., 2020). In the context of CompPath, explainability is defined as making the DL decisions understandable to pathologists (Holzinger et al., 2017). To this end, the qualitative evaluation of explainers' explanations by pathologists is the candid measure. However, it requires evaluations by task-specific expert pathologists, which is subjective, time-consuming, cum-

bersome, and expensive. Additionally, though the explanations are intuitive, they do not relate to pathologist-understandable terminologies, *e.g.*, "How big are the important nuclei?", "How irregular are their shape?" etc., which toughens the comprehensive analysis. These bottlenecks undermine not only any qualitative assessment but also quantitative metrics requiring user interactions (Mohseni et al., 2018). Furthermore, expressing the quantitative metrics in user-understandable terminologies with the appropriate *units of explanations* (Arrieta et al., 2020) is fundamental to achieve interpretability (Doshi-Velez and Kim, 2017; Nguyen and Rodriguez Martinez, 2020). Moreover, graph explainers usually intrinsically maintain high-*fidelity*, *e.g.*, GNNEXPLAINER (Ying et al., 2019) produces an explanation to match the GNN's prediction on the original graph. As a consequence, ensuring explainer *fidelity* (Mohseni et al., 2018; Pope et al., 2019; Ribeiro et al., 2016; Dhurandhar et al., 2017; Samek et al., 2017; Hoffman et al., 2018), while imperative, is not sufficient to characterize the explanation quality.

In this chapter, we present a set of novel user-independent quantitative metrics expressing pathologically-understandable *concepts*. The proposed metrics are based on class separability statistics using such *concepts*. They are also applicable in other domains by incorporating domain-specific prior knowledge. We use the proposed metrics to evaluate three types of graph-explainers, (i) graph pruning: GNNEXPLAINER (Ying et al., 2019; Jaume et al., 2020), (ii) gradient-based saliency: GRAPHGRAD-CAM (Selvaraju et al., 2017; Pope et al., 2019), GRAPHGRAD-CAM++ (Chattopadhay et al., 2018), (iii) layer-wise relevance propagation: GRAPHLRP (Bach et al., 2015; Montavon et al., 2015; Schwarzenberg et al., 2019), for explaining cell-graphs (Gunduz et al., 2004) towards the task of breast cancer subtyping. Figure 6.1 exemplifies a graph explanation derived from the GRAPHGRAD-CAM feature attribution method. The specific contributions presented in this chapter are:

- A set of novel quantitative metrics based on the statistics of class separability using domain-specific *concepts* to characterize graph explainability techniques. To the best of our knowledge, our metrics are the first of their kind to quantify explainability based on domain-understandable terminologies;
- Explainability in computational pathology using pathologically intuitive entity-graphs;
- Extensive qualitative and quantitative assessment of various graph explainability techniques in computational pathology, with a validation of the findings by expert pathologists.

6.2 Related work

Explainability is an integral part of pathological diagnosis. Though DL solutions have achieved remarkable diagnostic performance, their lack of explainability is unacceptable in the medical community (Tizhoosh and Pantanowitz, 2018). Recent studies have proposed visual explanations (Hagele et al., 2020) and salient regions (Bruno et al., 2017; Hagele et al., 2020)

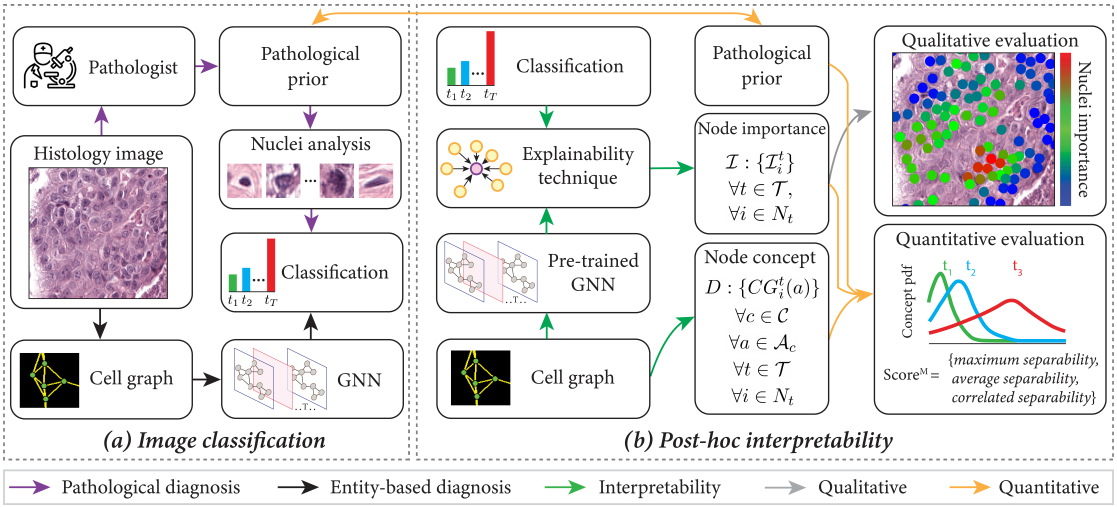


Figure 6.2 – Overview of the proposed framework. (a) presents pathologist, and entity-based (cell-graph + GNN) diagnosis of a histology image. (b) presents nuclei-level pathologically relevant *concept* measure D, a post-hoc graph explainability technique to derive nuclei-level importance $\mathscr{I}$ for *concepts* $\mathscr{C}$, measurable *attributes* $\mathscr{A}_c$, and classes $\mathscr{T}$. D, $\mathscr{I}$ and prior pathological knowledge defining *concepts*' relevance are utilized to propose a novel set of quantitative metrics to evaluate the explainer quality in pathologist-understandable terms.

using feature-attribution techniques (Selvaraju et al., 2017; Chattopadhay et al., 2018). Differently, concept-attribution technique (Graziani et al., 2020) evaluates the sensitivity of network output w.r.to quantifiable image-level pathological *concepts* in patches. Although such explanations are pathologist-friendly, image-level *concepts* are neither fit nor meaningful for real-world large histology images that contain many localized concepts. Furthermore, attention-based learning (Lu et al., 2021b), and multimodal mapping between image and diagnostic report (Zhang et al., 2019) are devised to localize network attention. All the aforementioned techniques are based on pixel- and patch-level processing, thus ignoring the notion of biological entity which makes them difficult to interpret by pathologists. Separately, the earlier stated entity graph-based processing provides an intuitive platform for pathologists. However, research on explainability and visualization using entity-graphs has been scarce: CGC-Net (Zhou et al., 2019a) analyzes cluster assignment of nodes in CG to group them according to their appearance and tissue types. Robust spatial filtering Sureka et al. (2020) utilizes an attention-based GNN and node occlusion to highlight cell contributions. No previous work has comprehensively analyzed and quantified graph explainers in computational pathology while expressing explanations in a pathologist-understandable form to the best of our knowledge. This gap between the existing and desired explainability of DL outputs in digital pathology motivates our work herein.

6.3 Method

In this section, we present the entity-graph processing, the set of considered explainability methods, and our proposed evaluation metrics. First, we transform a histology TRoI into a *biological entity-graph*. Second, we introduce a "black-box" GNN that maps the *entity-graph* to a corresponding class label. Third, we employ a post-hoc graph explainer to generate explanations. Finally, we perform a qualitative and quantitative assessments of the generated explanations. An overview of the methodology is shown in Figure 6.2.

6.3.1 Entity-graph construction

Following the notation introduced in Chapter 1, we define an attributed entity-graph $G := (V, E, H)$ as a set of nodes V, edges E, and node attributes $H \in \mathbb{R}^{|V| \times d}$. d denotes the number of attributes per node, and $|.|$ denotes set cardinality. The graph topology is defined by the adjacency matrix, $A \in \mathbb{R}^{|V| \times |V|}$, where $A_{u,v} = 1$ if $e_{uv} \in E$. We denote the neighborhood of a node $v \in V$ as $\mathcal{N}(v) := \{u \in V \mid v \in V,\ e_{uv} \in E\}$. Finally, we denote a set of graphs as $\mathcal{G}$.

Our methodology begins with transforming TRoIs into entity-graphs. It ensures the method's inputs are pathologically interpretable, as the inputs consist of biologically-defined objects that pathologists can directly *relate-to* and *reason-with*. Thus, image-to-graph conversion moves from an *uninterpretable* to *interpretable* input space. In this chapter, we consider cells as entities, thereby transforming TRoIs into cell-graphs (CGs). A CG nodes and edges capture the morphology of cells and cellular interactions. A CG topology acquires both tissue micro and macro-environment, which is crucial for characterizing cancer subtypes.

The CG construction is based on the work presented in Chapter 5. We herein provide the key steps - the reader may refer to Chapter 5 for a thorough description. First, we detect nuclei in a TRoI at 40× magnification using Hover-Net (Graham et al., 2019a), a nuclei segmentation algorithm pre-trained on MoNuSeg (Kumar et al., 2017). We process patches of size 72×72 pixels around the nuclei by ResNet34 (He et al., 2016) pre-trained on ImageNet (Deng et al., 2009b) to produce nuclei visual attributes. We further concatenate spatial attributes obtained by min-max normalization of nuclei centroids by the TRoI dimension. The nuclei and their attributes (visual and spatial) define the nodes and node attributes of the CG, respectively. We construct the CG topology by employing a thresholded k-NN algorithm. We set $k = 5$, and prune the edges longer than 50 pixels (12.5 μm). The CG-topology encodes how likely two nearby nuclei will interact (Francis and Palsson, 1997). A CG example is presented in Figure 6.1.

6.3.2 Entity graph learning

Given the set of CGs $\mathcal{G}$, the aim is to infer the corresponding cancer subtypes. This is a graph classification task that can be accurately modeled with a GNN. In this work, we use a flavor

of Graph Isomorphism Network (GIN) Xu et al. (2019b), that uses *mean* and a *multi-layer perceptron* (MLP) in the *aggregation* and *update* step respectively. Formally, a layer is defined as,

$$h(v)^{(t+1)} = \text{MLP}^{(t)}\left(h(v)^{(t)} + \frac{1}{|\mathcal{N}(v)|}\sum_{u\in\mathcal{N}(v)} h(u)^{(t)}\right) \tag{6.1}$$

where $h(v)$ denotes features of node v, and $t \in \{1,...,T\}$. Our GNN consists of 3-GIN layers, with each layer including a 2-layer MLP. The dimension of latent node embeddings is fixed to 64 for all layers. We use a *mean* operator as *readout step*, and feed the graph embedding to a 2-layer MLP classifier. The GNN is trained end-to-end by minimizing the cross-entropy loss between the predicted logits and the target cancer subtypes. We emphasize that the entity-based processing follows a pathologist's diagnostic procedure that identifies diagnostically relevant nuclei and analyzes cellular morphology and topology in a TRoI, as shown in Figure 6.2.

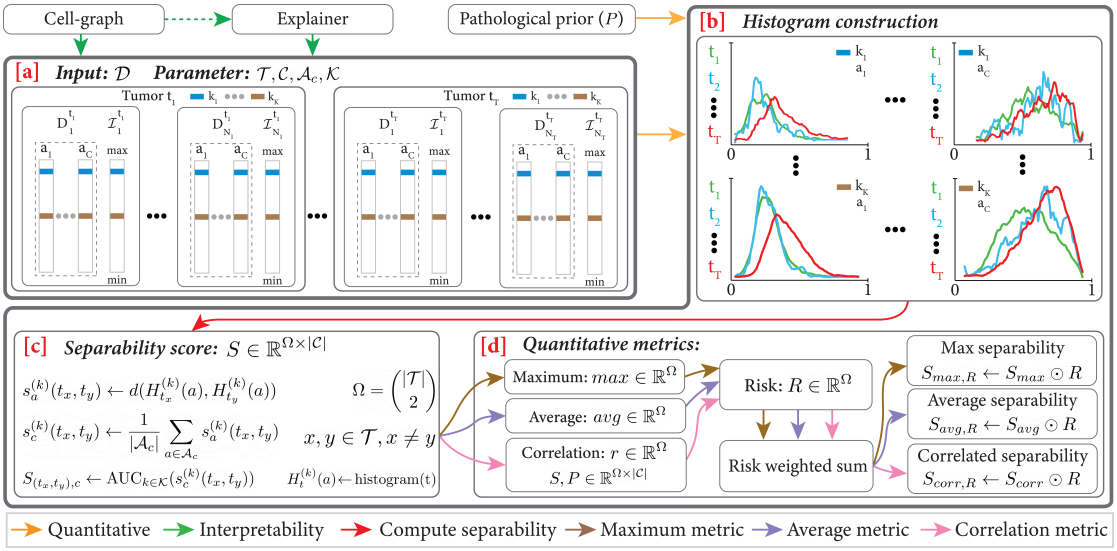


Figure 6.3 – Overview of the proposed quantitative assessment pipeline. (a) presents the input CG dataset $\mathscr{D}$, the set of *concepts* $\mathscr{C}$ and corresponding measurable *attributes* $\mathscr{A}_c$, the set of classes $\mathscr{T}$, and the set of importance thresholds $\mathscr{K}$. For simplicity $|\mathscr{A}_c| = 1, \forall c \in \mathscr{C}$ in this figure. (b) shows histogram probability densities for $\forall a \in \mathscr{A}_c, \forall k \in \mathscr{K}, \forall t \in \mathscr{T}$. (c) displays the algorithm for computing the class separability scores S. (d) presents the algorithm for computing the proposed class separability-based risk-weighted quantitative metrics.

6.3.3 Post-hoc graph explainer

We generate an explanation per entity-graph by employing post-hoc graph explainers. The explanations allow to evaluate the pathological relevance of black-box neural network reasoning. Specifically, we aim to evaluate the agreement between the pathologically relevant set of nuclei in a TRoI, and the explainer identified set of important nuclei, *i.e.*, the set of nuclei driving the prediction in a given cell-graph. In this work, we consider three types of graph

explainers for explaining CGs, which follow similar operational setting: (1) the input data are attributed graphs, (2) a GNN is trained *a priori* to classify the input entity-graph, and (3) each data point can be inferred independently to produce an explanation.

We succinctly present the graph explainers in the following sections. A detailed mathematical description was presented in Chapter 3.

GRAPHLRP

Layerwise relevance propagation (LRP) (Bach et al., 2015) propagates the output logits backward in the network using a set of propagation rules to quantify the contribution of each input pixel. Specifically, LRP assigns an importance score to each neuron such that the output logit relevance is preserved across layers. While initially developed for explaining fully-connected layers, LRP can be extended to GNN by treating the GNN *aggregation step* as a fully connected layer that projects the graph adjacency matrix on the node attributes as in (Schwarzenberg et al., 2019). LRP outputs per-node importance scores.

GRAPHGRAD-CAM

GRAD-CAM (Selvaraju et al., 2017) is a feature attribution approach designed for explaining CNNs operating on images. It produces class activation explanation following two steps. First, it assigns weights to each channel of a convolutional layer t by computing the gradient of the targeted output logit w.r.to each channel in layer t. Second, importance of the input elements are computed by the weighted combination of the forward activations at each channel in layer t. The extension to GNNs is straightforward Pope et al. (2019), and only requires to compute the gradient of the predicted logits w.r.to a GNN layer. Following prior work Pope et al. (2019), we take the average of node-level importance-maps obtained from all the GNN layers $t \in \{1, ..., T\}$ to produce smooth per-node importance scores.

GRAPHGRAD-CAM++

GRAD-CAM++ (Chattopadhay et al., 2018) is an increment on GRAD-CAM that includes spatial contributions into the channel-wise weight computation of a convolutional layer. The spatial locations in a convolutional layer are analogous to the size of the graph in a GNN layer. With this additional consideration, we derived an extension of GRAD-CAM++ applicable to graph-structured data (see Appendix A).

GNNEXPLAINER

GNNEXPLAINER (Ying et al., 2019; Jaume et al., 2020) is a graph pruning approach that aims to find a compact sub-graph $G_s \subset G$ such that the mutual information between G_s and the GNN prediction of G is maximized. The sub-graph G_s is regarded as the explanation for the

input graph G. GNNEXPLAINER can be seen as a feature attribution technique with binarized node importance scores. To address the combinatorial nature of finding G_s, GNNEXPLAINER formulates it as an optimization problem that learns a mask to activate or deactivate parts of the graph. Jaume et al. (2020) reformulates the initial approach in Ying et al. (2019) to learn a mask over the nodes instead of edges. The approach in Jaume et al. (2020) is better suited for pathology as the nodes, *i.e.*, biological entities, are more intuitive and substantial for disease diagnosis than the edges that remain heuristically-defined. The optimization for an entity graph results in per-node importance.

RANDOM

We further introduce a RANDOM explainer to assess a lower bound per quantitative metric. The RANDOM baseline is simply implemented by using a *random* nuclei selection.

6.3.4 Quantitative metrics for graph explainability

In the presence of several graph explainers producing distinct explanations for an input, it is primordial to discern the explainer that produces the most pathologically-aligned explanations. Considering the limitations of existing qualitative and quantitative measures presented at the beginning of the chapter, we propose a novel set of quantitative metrics based on class separability statistics using pathologically relevant *concepts*. Intuitively, a good explainer should emphasize the relevant *concepts* that maximize the class separation. Details of the metric evaluations are presented as follows.

Input: A graph explainer outputs an explanation, *i.e.*, node-level *importance* scores $\mathscr{I}$, for an input CG. To quantify a *concept* $c \in \mathscr{C}$, $\mathscr{C}$ denoting the set of *concepts*, we measure a set of nuclear *attributes* $a \in \mathscr{A}_c$ for each nucleus in CG. For instance, in order to represent the concept c = *nuclear shape*, we measure the attribute set $\mathscr{A}_c$ = {*perimeter, roughness, eccentricity, circularity*}. We create a dataset $\mathscr{D} = \bigcup_{t \in \mathscr{T}} \mathscr{D}_t$, $\mathscr{T}$ denoting the set of cancer subtypes. We define $\mathscr{D}_t := \{(D_i^t, \mathscr{I}_i^t) | i = 1, \ldots, N_t\} \forall\, t \in \mathscr{T}$, where N_t is the number of CGs for tumor type t. $\mathscr{I}_i^t$ and D_i^t are, respectively, the sorted importance matrix for a CG indexed by i and corresponding node-level attribute matrix. To perform inter-concept comparisons, we conduct *attribute*-wise normalization across all samples $D_i^t\ \forall\, t, i$. In order to compare different explainers, we conduct CG-wise normalization of importance scores $\mathscr{I}$. The structure of input the generated dataset $\mathscr{D}$ is presented in Figure 6.2(a).

Note that the notion of important nuclei vary (1) per-CG since the number of nodes vary across CGs, and (2) per-explainer. Hence, selecting a *fixed* number of important nuclei per-CG and per-explainer is not meaningful. To overcome this issue, we assess different number of important nuclei $k \in \mathscr{K}$, selected based on node importances, per-CG and per-explainer. In the following sections we will show how to aggregate the results for a given explainer.

Histogram construction: Given the input dataset $\mathscr{D}$, and $\mathscr{K}, \mathscr{C}, \mathscr{A}_c, \mathscr{T}$, we apply a threshold

$k \in \mathscr{K}$ on $\mathscr{I}_i^t, \forall t \in \mathscr{T}, \forall i \in N_t$ to select the CG-wise most important nuclei. The cancer subtype-wise selected set of nuclei data from $\mathscr{D}$ are used to construct histograms $H_t^{(k)}(a), \forall a \in \mathscr{A}_c$, $\forall c \in \mathscr{C}$ and $\forall t \in \mathscr{T}$. For a given histogram $H_t^{(k)}(a)$, bin-edges are defined by quantizing the complete range of *attributes a, i.e.*, $\mathscr{D}(a)$, by a fixed step size. We further convert each $H_t^{(k)}(a)$ into a probability density function. Similarly, sets of histograms are constructed by applying different thresholds $k \in \mathscr{K}$. Examples of histograms are shown in Figure 6.2(b).

Separability scores (S): Given two classes $t_x, t_y \in \mathscr{T}$ and corresponding probability density functions $H_{t_x}^{(k)}(a)$ and $H_{t_y}^{(k)}(a)$, we compute a *class separability* score $s_a^{(k)}(t_x, t_y)$ based on optimal transport as the Wasserstein distance between the two density functions. We average $s_a^{(k)}(t_x, t_y)$ over all the attributes $a \in \mathscr{A}_c$ to obtain a unique score $s_c^{(k)}(t_x, t_y)$ for *concept c* and threshold k. Finally, we compute the area-under-the-curve (AUC) over the threshold range $\mathscr{K}$ to get the aggregated class separability scores $S_{(t_x,t_y),c}$ for a *concept c*. The class separability score indicates the significance of a *concept c* for the purpose of separating the classes t_x and t_y. Thus, separability scores can be used to compare different *concepts* and to identify relevant ones for differentiating t_x and t_y. A pseudo-algorithm is presented in Algorithm 1, and illustrated in Figure 6.2(c). Finally, a separability matrix $S \in \mathbb{R}^{\Omega \times |\mathscr{C}|}$ is built by computing class separability scores for all pairs of classes, *i.e.*, $\forall\ (t_x, t_y) \in \Omega := \binom{|\mathscr{T}|}{2}$ and $\forall c \in \mathscr{C}$.

Statistics of separability scores: Since the notion of explainability is not uniquely defined, we define multiple metrics highlighting different facets and providing different insights. We compute three separability statistics $\forall (t_x, t_y) \in \Omega$ using S as given in Equation (6.2), *i.e.*, (1) *maximum*: the utmost separability, (2) *average*: the expected separability. These two metrics encode (model+explainer)'s focus, *i.e.*, "how much the black-box model implicitly uses the *concepts* for class separability?" (3) *correlation*: encodes the agreement between (model+explainer)'s focus and pathological prior P. $P \in \mathbb{R}^{\Omega \times |\mathscr{C}|}$ signifies the relevance $\forall c \in \mathscr{C}$ for differentiating $(t_x, t_y) \in \Omega$, *e.g.*, nuclear *size* is highly relevant for classifying benign and malignant tumor as important nuclei in malignant are larger than important nuclei in benign. Specifically, the metrics are defined as:

$$
\begin{aligned}
s_{\max}(t_x, t_y) &= \max_{c \in \mathscr{C}} S_{(t_x,t_y),c} \\
s_{\text{avg}}(t_x, t_y) &= \frac{1}{|C|} \sum_{c \in \mathscr{C}} S_{(t_x,t_y),c} \\
s_{\text{corr}}(t_x, t_y) &= \rho(S_{(t_x,t_y),c=1,..,|\mathscr{C}|}, P_{(t_x,t_y),c=1,..,|\mathscr{C}|})
\end{aligned}
\tag{6.2}
$$

where ρ denotes Pearson correlation. $s_{\max}$, $s_{avg} \in [0,\infty)$ show separation between unnormalized class-histograms; and $s_{corr} \in [-1, 1]$ shows agreement between S and P. We build $S_{\max}$, S_{avg} and S_{corr} by computing Equation (6.2) for all pairs of classes $\forall (t_x, t_y) \in \Omega$. Metrics' complementary may lead to relevant *concepts* different to pathological understanding.

Risk: We further introduce the notion of risk as a weight to indicate the cost of misclassifying a sample of class t_x, erroneously as class t_y (Thai-Nghe et al., 2010; He and Ma, 2013). Indeed, misclassifying a malignant tumor as a benign tumor is riskier than misclassifying it as an

atypical tumor. Thus, we construct a risk vector $R \in \mathbb{R}^{\Omega}$. In this work, each entry in R defines the symmetric risk of differentiating t_x from t_y measured as the number of class-hops needed for a tumor type to progress from t_x to t_y.

Metrics: Finally, we propose three quantitative metrics based on class separability to assess an explainer quality. The metrics are computed as the risk weighted sum of the statistics of separability scores. Namely, we define:

- the *maximum separability* $S_{\max,R} := S_{\max} \odot R$;
- the *average separability* $S_{\text{avg},R} := S_{\text{avg}} \odot R$;
- and the *correlated separability* $S_{\text{corr},R} := S_{\text{corr}} \odot R$, where $\odot$ defines the Hadamard product.

The first two metrics are pathologist-independent, and the third metric requires expert pathologists to impart the domain knowledge in the form of pathological prior P. Such prior can be defined individually by a pathologist or collectively by consensus of several pathologists, and it is independent of the algorithm generated explanations.

Algorithm 1: Class separability computation.

Input: $\mathscr{D} = \{(D_i^t, \mathscr{I}_i^t)\}, t \in \mathscr{T}, i \in N_t$
Parameter: $\mathscr{T}, \mathscr{C}, \mathscr{A}_c, \mathscr{K}$
Result: $S \in \mathbb{R}^{\binom{|\mathscr{T}|}{2} \times |\mathscr{C}|}$
for $c \in \mathscr{C}$ **do** // go over concepts
  for $k \in \mathscr{K}$ **do** // go over nuclei thresh
    for $a \in \mathscr{A}_c$ **do** // go over attributes
      for $t \in \mathscr{T}$ **do** // go over classes
        var $\leftarrow D_i^t(a)[:k]$ // sorted I_i^t
        $H_t^{(k)}(a) \leftarrow$ histogram(var)
      for $(t_x, t_y) \in \binom{|\mathscr{T}|}{2}$ **do** // go over class pairs
        $s_a^{(k)}(t_x, t_y) \leftarrow d(H_{t_x}^{(k)}(a), H_{t_y}^{(k)}(a))$
    $s_c^{(k)}(t_x, t_y) \leftarrow \frac{1}{|\mathscr{A}_c|} \sum_{a \in \mathscr{A}_c} s_a^{(k)}(t_x, t_y)$
  $S_{(t_x,t_y),c} \leftarrow \text{AUC}_{k \in \mathscr{K}}(s_c^{(k)}(t_x, t_y))$

6.3.5 Concepts and attributes

For cancer subtyping, relevant *concepts* are nuclear morphology and topology (Rajbongshi et al., 2018; Kashyap et al., 2018; Nguyen et al., 2017; Allison et al., 2016).

In this work, we focus on pathologically-understandable nuclear *concepts* $\mathscr{C}$ pertaining to nuclear morphology for breast cancer subtyping. To quantify each $c \in \mathscr{C}$, we use several

Concept ($\mathscr{C}$)	Attribute ($\mathscr{A}$)	Computation	Benign	Atypical	Malignant
Size	Area	$A(x)$	Small	Small-Medium	Medium-Large
Shape	Perimeter	$P(x)$	Smooth	Mild irregular	Irregular
	Roughness	$\frac{P_{\text{ConvHull}}(x)}{P(x)}$			
	Eccentricity	$\frac{a_{\text{minor}}(x)}{a_{\text{major}}(x)}$			
	Circularity	$\frac{4\pi A(x))}{P(x)^2}$			
Shape variation	Shape factor	$\frac{4\pi A(x)}{P^2_{\text{ConvHull}}}$	Monomorphic	Monomorphic	Pleomorphic
Spacing	Mean spacing	$\text{mean}(d_y \mid y \in \text{kNN(x)})$	Evenly crowded	Evenly spaced	Variable
	Std spacing	$\text{std}(d_y \mid y \in \text{kNN(x)})$			
Chromatin	GLCM dissimilarity	$\sum_i \sum_j \lvert i-j \rvert p(i,j)$	Light euchromatic	Hyperchromatic	Vesicular
	GLCM contrast	$\sum_i \sum_j (i-j)^2 p(i,j)$			
	GLCM homogenity	$\sum_i \sum_j \frac{p(i,j)}{1+(i-j)^2}$			
	GLCM ASM	$\sum_i \sum_j p(i,j)^2$			
	GLCM entropy	$-\sum_i \sum_j p(i,j) \log(p(i,j))$			
	GLCM variance	$\sum_i \sum_j (i-\mu_i)^2 p(i,j)$ with $\mu_i = \sum_i \sum_j i p(i,j)$			

Table 6.1 – Pathologically-understandable nuclear *concepts*, corresponding measurable *attributes*, and computations are shown in Columns 1, 2, 3, respectively. The expected *concept* behavior for three breast cancer subtypes is shown in Columns 4, 5, 6, respectively.

measurable *attributes* $\mathscr{A}_c$. Table 6.1 presents the list of *concepts* and corresponding *attributes* used to perform the proposed quantitative analysis in this work. Also, Table 6.1 includes the class-wise expected criteria for each *concept*. The *attributes* of the nuclei in a TRoI are computed as presented in Table 6.1. It uses the TRoI and corresponding nuclei segmentation map, denoted as I_{seg}. Area of a nucleus x, denoted as $A(x)$, is defined as the number of pixels belonging to x in I_{seg}. $P(x)$, the perimeter of x, is measured as the contour length of x in I_{seg}. $P_{\text{ConvHull}}(x)$, the convex hull perimeter of x, is defined as the contour length of convex hull induced by x in I_{seg}. The major and minor axis of x, noted as $a_{\text{major}}(x)$ and $a_{\text{minor}}(x)$, are the longest diameter of x and the longest line segment perpendicular to $a_{\text{major}}(x)$, respectively. The chromatin *attributes* are computed from the normalized gray level co-occurrence matrix (GLCM) (Haralick et al., 1973), which captures the probability distribution of co-occurring gray values in x. In all our experiments, we select $\mathscr{K} = \{5, 10, ..., 50\}$ nuclei per CG.

6.4 Results

This section describes the analysis of CG explainability for breast cancer subtyping. We evaluate three types of graph explainers and quantitatively analyze the explainer quality using the proposed class separability metrics.

6.4.1 Dataset

We experiment on the BRACS dataset, introduced in Chapter 5. The dataset consists of 4391 TRoIs at 40× resolution from 325 H&E stained breast carcinoma whole-slides. In order to simplify the analysis, the TRoIs are grouped under three classes as, (1) Benign (B): normal (N), benign (B) and usual ductal hyperplasia (UDH), (2) Atypical (A): flat epithelial atypia (FEA) and atypical ductal hyperplasia (ADH), and (3) Malignant (M): ductal carcinoma *in situ* (DCIS) and invasive (I).

6.4.2 Training

We conducted our experiments using PyTorch (Paszke et al., 2019) and DGL (Wang et al., 2019a). The GNN architecture for CG classification is presented in Section 6.3.2. The CG classifier was trained for 100 epochs using the Adam optimizer (Kingma and Ba, 2015), 10^{-3} learning rate and 16 batch size. The best CG-classifier achieved 74.2% weighted F1-score on the test set for the three-class classification. The average time for processing a 1K×1K TRoI on a NVIDIA P100 GPU is 2s for the CG generation and 0.01s to run the GNN inference.

6.4.3 Qualitative assessment

Graph explainer qualitative comparison

Figure 6.4 presents explanations, *i.e.*, nuclei importance maps, from four studied graph explainers. We observe that GRAPHGRAD-CAM and GRAPHGRAD-CAM++ produce similar importance maps. The GNNEXPLAINER generates almost binarized nuclei importances. Interestingly, the gradient and pruning-based techniques consistently highlight similar regions. Indeed, the approaches focus on relevant epithelial region and discard stromal nuclei and lymphocytes outside the glands. Differently, GRAPHLRP produces less interpretable maps through high spatial localization (Figure 6.4(d)) or less spatial localization (Figure 6.4(h,l)).

GNNEXPLAINER qualitative analysis

As GNNEXPLAINER provides (almost) binarized importance weights, we can easily visualize all the important nuclei. By modifying the class assignment of the BRACS dataset, we can define three different scenarios of increasing complexity: (i) a 2-class problem: benign (N+B) and malignant (D+I) categories, a (ii) a 3-class problem: benign (N+B), atypical (A), and malignant (D+I) categories, and (iii) a 5-class problem: normal (N), benign (B+UDH), atypical (ADH+FEA), carcinoma *in situ* (DCIS) and invasive (I). These scenarios allow to study the relation between the task complexity and the generated explanations. Combining the CG explanations in Figure 6.5 and the nuclei types annotation in Figure 6.6, we infer that the explanations retain relevant tumor epithelial nuclei for DCIS diagnosis. For the 2-class scenario, the CG includes tumor nuclei in the central region of the gland. Few tumor nuclei are sufficient to

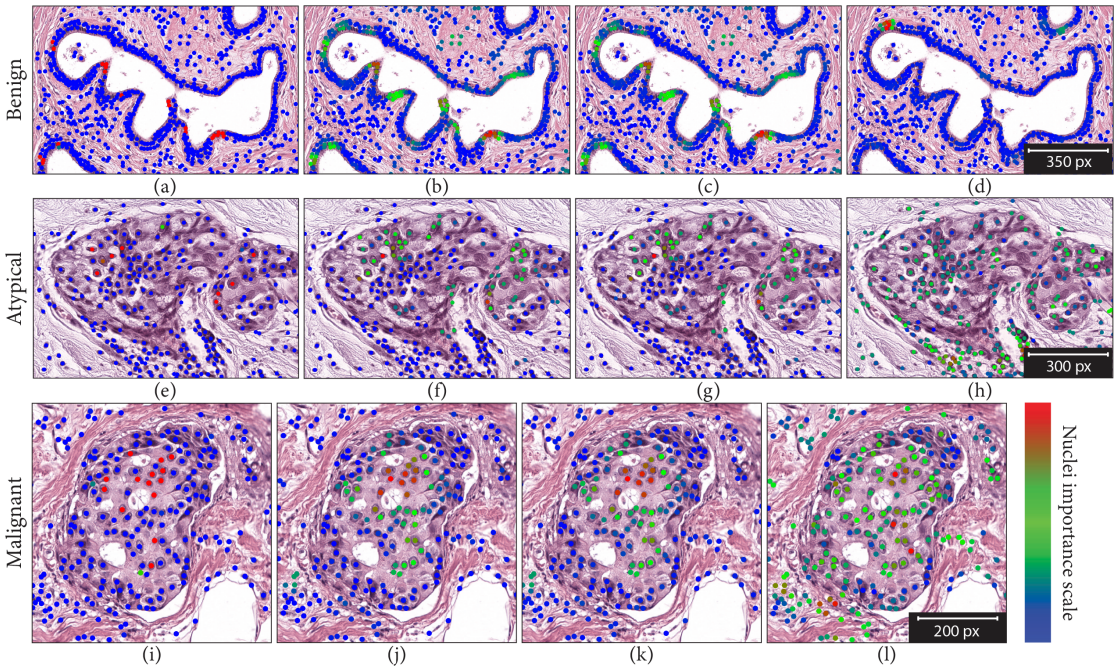


Figure 6.4 – Qualitative results. The rows represent the cancer subtypes, *i.e.*, Benign, Atypical and Malignant, and the columns represent the graph explainability techniques, *i.e.*, GNNEXPLAINER, GRAPHGRAD-CAM, GRAPHGRAD-CAM++, and GRAPHLRP. Nuclei-level importance ranges from blue (the least important) to red (the most important).

differentiate (DCIS) from (N+B). For the 3-class scenario, the CG includes more tumor nuclei in the central region and the periphery of the gland and does not consider atypical nuclei. This pattern differentiates (DCIS) from (A). For the 5-class scenario, the CG includes more tumor nuclei distributed within and around the gland, and some lymphocytes around the gland. The CG also includes more cellular interactions to identify a large cluster of tumor nuclei. Pathologically this behavior differentiates (DCIS) from (I) which has small clusters of tumor nuclei scattered throughout the TRoI. Additionally, the retained tumor nuclei and their interactions are consistent with increasing task complexity, *i.e.*, all the important nuclei selected in the 2-class scenario are kept in the 3-class and 5-class ones. While consistent, this analysis does not provide insight into the underlying mechanism that results in the nuclei selection.

Qualitative visual assessment of Figure 6.4 and Figure 6.5 conclude that, (i) *fidelity* preserving explainers result differently based on the underlying explainability technique, (ii) high *fidelity* does not guarantee straightforward pathologist-understandable explanations, (iii) qualitative assessment cannot rigorously compare explainers' quality, and (iv) large-scale pathological evaluation is inevitable to rigorously rank the explainers.

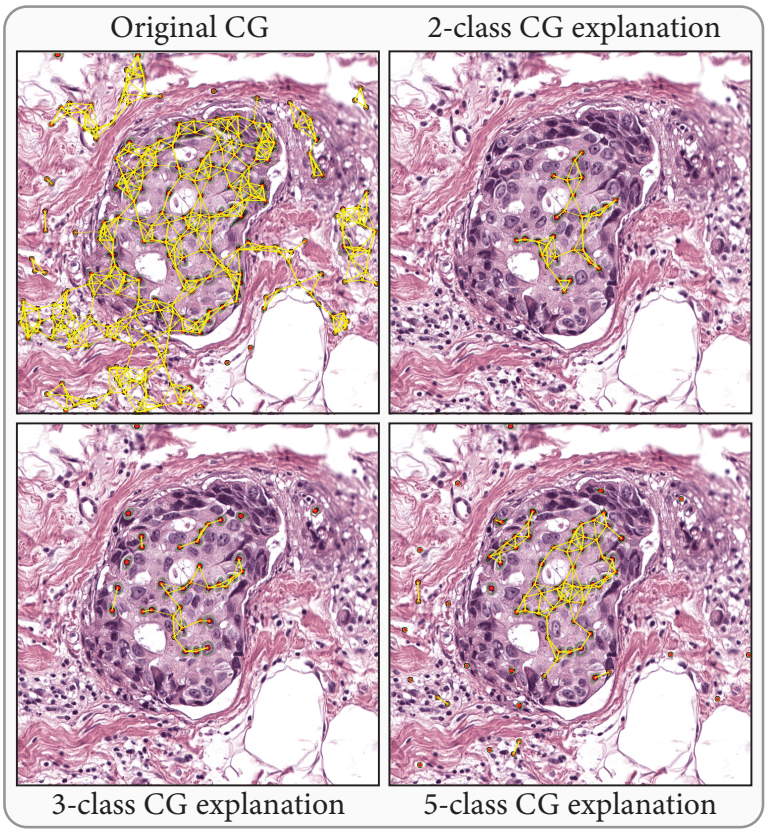


Figure 6.5 – Qualitative comparison of original CG and GNNEXPLAINER CGs for 2, 3 and 5-class scenarios for a DCIS TRoI.

6.4.4 Quantitative analysis

Histogram analysis

Histogram construction is a key component in the proposed quantitative metrics. Figure 6.7 presents per-class histograms for each explainer and the best *attribute* per *concept*. We set the importance threshold to $k = 25$, *i.e.*, for each TRoI, we select 25 nuclei with the highest node importance scores. The best *attribute* for a *concept* is the one with the highest average pair-wise class separability.

The row-wise observation exhibits that GNNEXPLAINER and GRAPHLRP provide, respectively, the maximum and the minimum pair-wise class separability. The histograms for a *concept* and for an explainer can be analyzed to assess the agreement between the selected important nuclei *concept*, and the expected *concept* behavior as presented in Table 6.1, for all the classes. For instance, nuclear *area* is expected to be higher for malignant TRoIs than benign ones. The *area* histograms for GNNEXPLAINER, GRAPHGRAD-CAM and GRAPHGRAD-CAM++ indicate that the important nuclei set in malignant TRoIs includes nuclei with higher area compared to benign TRoIs. Similarly, the important nuclei in malignant TRoIs are expected to be vesicular,

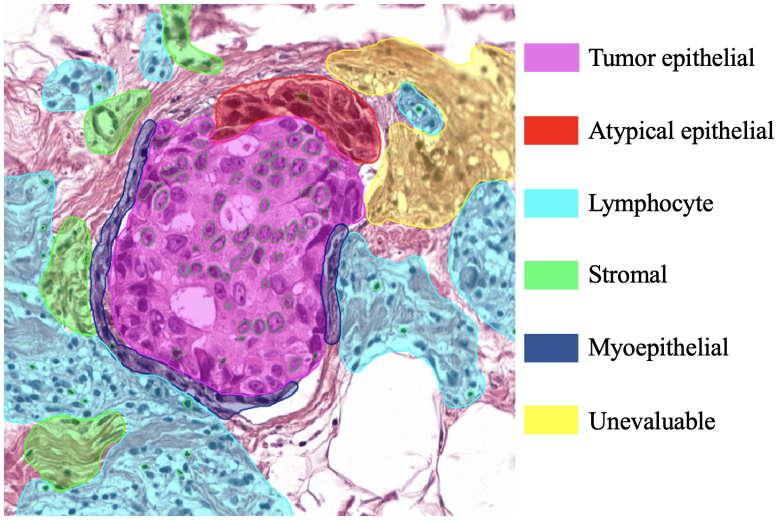


Figure 6.6 – Nuclei types annotation. Overlaid segmentation masks of nuclei from 5-class explanation in green.

i.e., high texture entropy, compared to light euchromatic, *i.e.*, moderate texture entropy, in benign TRoIs. The *chromaticity* histograms for GNNEXPLAINER, GRAPHGRAD-CAM and GRAPHGRAD-CAM++ display this behavior. Additionally, the histogram analysis can reveal the important *concepts* and important *attributes*. For instance, nuclear *density* proves to be the least important *concept* for differentiating the classes.

Influence of threshold value on separability scores

Multiple importance thresholds $\mathscr{K}$ are required to address the varying notion of important nuclei across different cell-graphs and explainers. Figure 6.8 presents the behavior of the pair-wise class separability for using various $k \in \mathscr{K} = \{5, 10, ..., 50\}$. For simplicity, we present the behavior for the best *attribute* per *concept*. In general, the pair-wise class separability is observed to decrease with decreasing k. Intuitively, decreasing k results in including more unimportant nuclei into the evaluation, thereby gradually decreasing the class separability.

The degree of agreement between the difference in the expected behavior per *concept* and the pair-wise class separability in Figure 6.8, for all pair-wise classifications and various $k \in \mathscr{K}$ can be used to assess the explainer's quality. For instance, according to Table 6.1, the difference in the expected nuclear *size* can be considered as benign–atypical < benign–malignant, and atypical–malignant < benign–malignant. GNNEXPLAINER, GRAPHGRAD-CAM and GRAPHGRAD-CAM++ display these behaviors $\forall k \in \mathscr{K}$. GNNEXPLAINER provides the highest class separability in each pair-wise classification, thus proving to be the best explainer pertaining to *size concept*. Detailed inspection of Figure 6.8 shows that all the differences in the expected behavior, per *concept* for all pair-wise classifications, is inline with the *concept*-wise expected behavior in Table 6.1, $\forall c \in \mathscr{C}$ and $\forall k \in \mathscr{K}$. Overall, GNNEXPLAINER is seen to be the

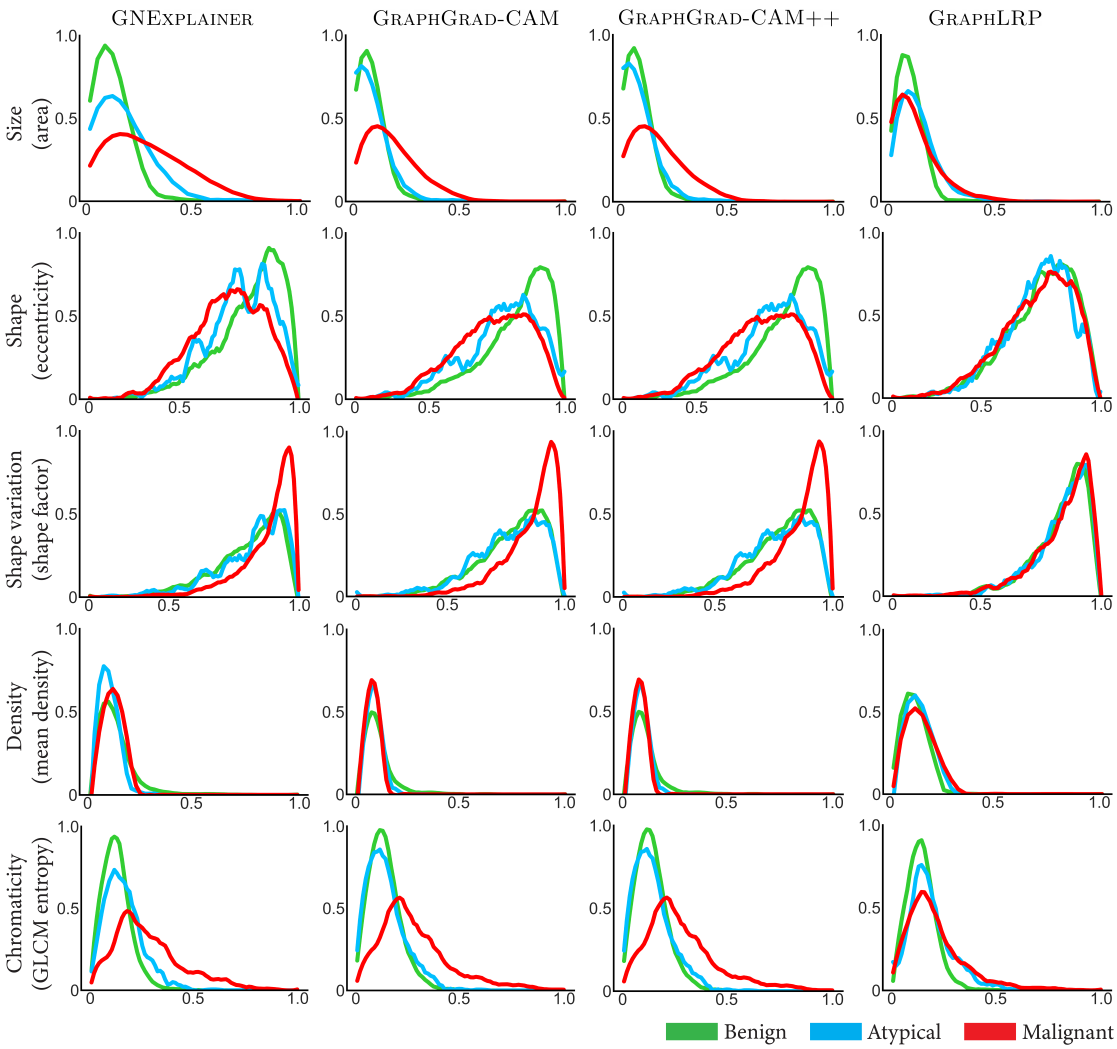


Figure 6.7 – Per-class histograms for different *concepts* across different graph explainers. For simplicity, histograms are presented for the best *attribute* per *concept* at fixed importance threshold $k = 25$.

best explainer as it agrees to the majority of the expected differences $\forall c \in \mathscr{C}$ for all pair-wise classifications, while providing high-class separability. Furthermore, *size* proves to be the most important *concept* that provides the maximum class separability across all pair-wise classifications.

Separability score analysis

Table 6.2 presents the statistics of pair-wise class separability and aggregated separability w/ and w/o risk to assess the studied explainers quantitatively. Also, for each class pair (t_x, t_y), we compute classification accuracy by using the CGs of type t_x, t_y.

Tasks (Ω)		B vs. A	B vs. M	A vs. M	B vs. A vs. M			
Accuracy (in %)		77.19	90.29	80.42	74.92			
Explainer	Metric $\forall\,(t_x, t_y) \in \Omega$ (↑)				Agg. Metric w/o Risk (↑)		Agg. Metric w/ Risk (↑)	
GNNEXPLAINER	$s_{\max}(t_x, t_y)$	**3.26**	**6.24**	**3.48**	$S_{\max}$	**12.98**	$S_{\max,R}$	**19.22**
GRAPHGRAD-CAM		1.24	4.41	3.36		9.01		13.42
GRAPHGRAD-CAM++		1.27	<u>4.42</u>	<u>3.40</u>		<u>9.09</u>		<u>13.51</u>
GRAPHLRP		<u>2.33</u>	2.46	1.28		6.07		8.53
RANDOM		1.02	1.26	1.11		3.39		4.65
GNNEXPLAINER	$s_{\text{avg}}(t_x, t_y)$	**1.54**	**2.78**	1.93	S_{avg}	**6.25**	$S_{\text{avg},R}$	**9.03**
GRAPHGRAD-CAM		1.15	2.57	<u>2.08</u>		5.80		8.37
GRAPHGRAD-CAM++		1.18	<u>2.58</u>	**2.09**		<u>5.85</u>		<u>8.43</u>
GRAPHLRP		<u>1.38</u>	1.59	1.47		4.44		6.03
RANDOM		1.05	1.00	0.95		3.00		4.00
GNNEXPLAINER	$s_{\text{corr}}(t_x, t_y)$	−0.02	0.36	0.38	S_{corr}	0.72	$S_{\text{corr},R}$	1.08
GRAPHGRAD-CAM		<u>−0.01</u>	<u>0.57</u>	<u>0.58</u>		<u>1.14</u>		<u>1.71</u>
GRAPHGRAD-CAM++		**−0.01**	**0.58**	**0.59**		**1.16**		**1.74**
GRAPHLRP		−0.15	−0.49	−0.23		−0.87		−1.36
RANDOM		−0.37	−0.31	−0.18		−0.86		−1.17

Table 6.2 – Quantitative assessment of graph explainers: GNNEXPLAINER, GRAPHGRAD-CAM, GRAPHGRAD-CAM++ and GRAPHLRP, using proposed *maximum, average*, and *correlated separability* metrics. Results are provided for each pair-wise breast subtyping tasks, and are aggregated w/o and w/ risk weighting, *i.e.*, $S_{\max}$ and $S_{\max,R}$. The first and second best values are indicated in **bold** and <u>underline</u>.

Noticeably, GNNEXPLAINER achieves the best *maximum* and *average separability* for the majority of pair-wise classes. GRAPHGRAD-CAM++ and GRAPHGRAD-CAM followed GNNEXPLAINER except for (B vs. A), where GRAPHLRP outperforms them. All explainers outperform the RANDOM baseline which conveys that the quality of the explainers' explanations are all better than random. Notably, GRAPHGRAD-CAM and GRAPHGRAD-CAM++ quantitatively perform very similarly, which is consistent with our qualitative analysis in Figure 6.4. Interestingly, a positive correlation is observed between pair-wise class accuracies and *average separability* for the explainers, *i.e.*, better classification leads to better *concept* separability, and thus produces better explanations. Further, the observation does not hold for the randomly generated explanations, which highlight undifferentiable *average concept* separability.

To obtain the pathological prior used to compute the *correlation separability*, we consulted three pathologists to rank the *concepts* in terms of their relevance for discriminating each pair of classes. For instance, given an atypical TRoI, we asked how important is nuclear *shape* to classify the TRoI as *not* benign and *not* malignant. To this end, a dataset of 100 TRoIs per class is employed. The ranked *concepts* are averaged across TRoIs belonging to a pair of classes, followed by a *min-max* normalization across all *concepts*. The outcome is a normalized prior matrix $P \in \mathbb{R}^{\Omega \times |\mathscr{C}|}$. We observe that GNNEXPLAINER, GRAPHGRAD-CAM and GRAPHGRAD-CAM++ have positive *correlated separability* for (B vs. M), (A vs. M), and nearly zero values for (B vs. A). It shows that the explanations for (B vs. M) and (A vs. M) bear similar relevance of *concepts* as the pathologists, and focus on a different relevance of *concepts* for (B vs. A). GRAPHGRAD-CAM++ has the best overall agreement at the *concept*-level with the pathologists, followed by GRAPHGRAD-CAM and GNNEXPLAINER. The RANDOM baseline agrees significantly worse than the three explainers, and GRAPHLRP has the least agreement.

Concept (Attributes) / Tasks (Ω)	B vs. A	B vs. M	A vs. M	w/o risk (↑)	w/ risk (↑)
Size	**3.26**	**6.24**	**3.47**	**12.97**	**19.21**
Shape	1.27	2.23	1.60	5.10	7.34
Shape variation	0.69	2.30	1.99	4.97	7.28
Density	1.01	0.80	0.52	2.33	3.14
Chromaticity	<u>1.44</u>	<u>2.31</u>	<u>2.07</u>	<u>5.82</u>	<u>8.13</u>
Average separability (↑)	1.54	2.78	1.93	6.25	9.03

Table 6.3 – Quantification of *concepts* for pair-wise and aggregated class separability in GNNEXPLAINER. The first and second best values are indicated in **bold** and <u>underline</u>. The per-*concept attributes* are presented in the first column. A comprehensive description of per-concept attributes is presented in Table 6.1.

Table 6.3 provides more insights by highlighting *concept*-level scores of GNNEXPLAINER. The nuclear *size* is the most relevant *concept*, followed by the *chromaticity* and the *shape variation*. Comparatively, the nuclear *density* is the least relevant *concept*.

6.5 Conclusion

In this chapter, we presented an approach for explaining black-box DL solutions in Comp-Path. We advocated for biological entity-based analysis instead of conventional pixel-wise analysis, thus providing an intuitive space for pathological understanding. We employed four graph explainability techniques, *i.e.*, graph pruning (GNNEXPLAINER), gradient-based saliency (GRAPHGRAD-CAM, GRAPHGRAD-CAM++) and layerwise relevance propagation (GRAPHLRP), to explain “black-box" GNNs. We proposed a novel set of user-independent quantitative metrics expressing pathologically-understandable *concepts* to evaluate the graph explainers, which relaxes the exhaustive qualitative assessment by expert pathologists. Our analysis concludes that the explainer bearing the best class separability in terms of *concepts* is GNNEXPLAINER, followed by GRAPHGRAD-CAM++ and GRAPHGRAD-CAM. GRAPHLRP is the worst explainer in this category while outperforming a randomly created explanation. We observed that the explainer quality is directly proportional to the GNN's classification performance for a pair of classes. Furthermore, GRAPHGRAD-CAM++ produces explanations that best agrees with the pathologists in terms of *concept* relevance, and objectively highlights the relevant set of *concepts*. Considering the expansion of entity graph-based processing, such as radiology, computation biology, satellite and natural images, graph explainability and their quantitative evaluation is crucial. The proposed method encompassing domain-specific user-understandable terminologies can potentially be of great use in this direction. It is a meta-method that is applicable to other domains and tasks by incorporating relevant entities and corresponding *concepts*. For instance, with entity-graph nodes denoting car/body parts in Stanford Cars (Krause et al., 2013)/ Human poses (Andriluka et al., 2014), and expert knowledge available on car-model/ activity, our method can infer relevant parts by quantifying their agreement with experts.

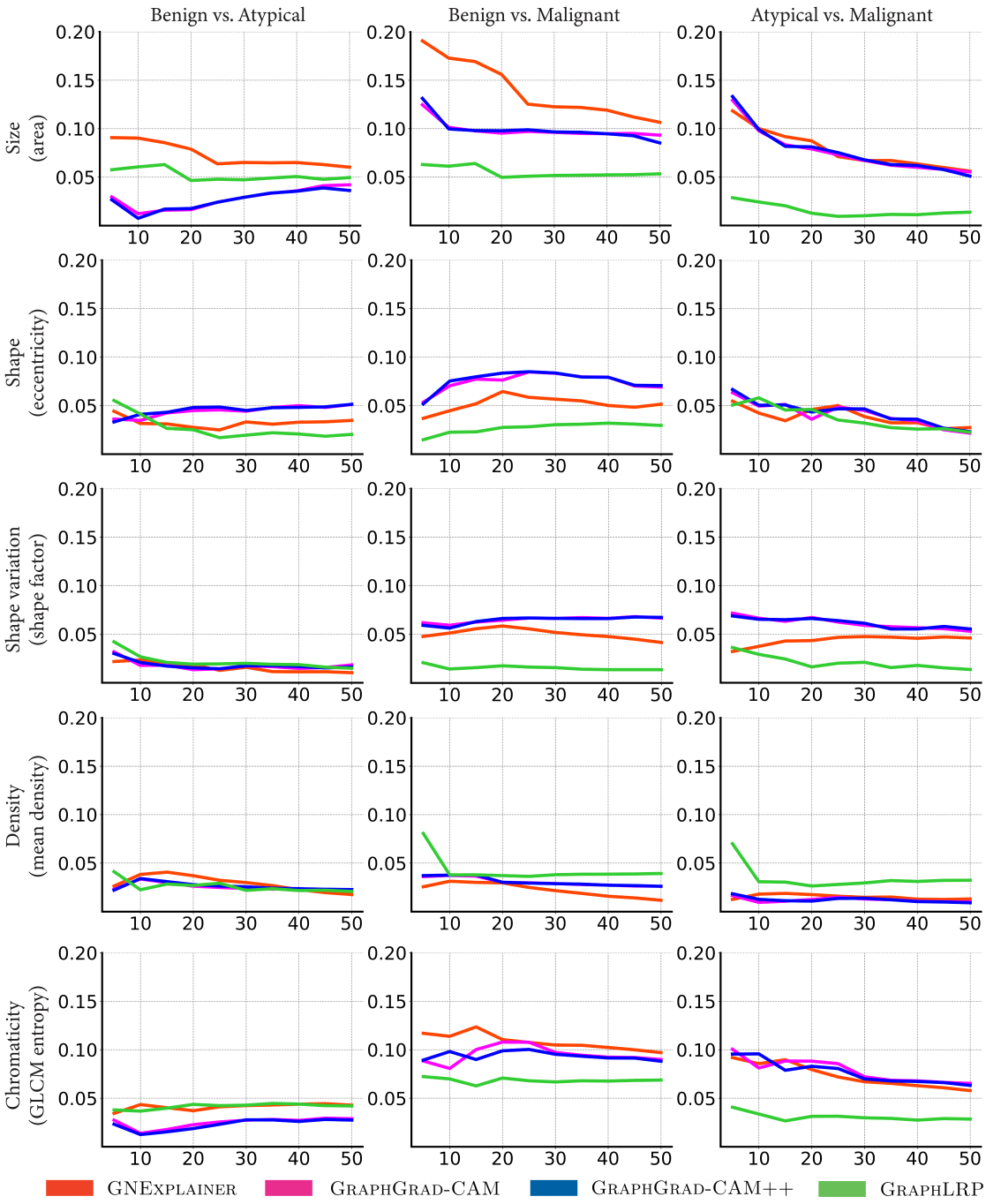


Figure 6.8 – Visualizing the variation of pair-wise class separability score (Y-axis) w.r.to various nuclei importance thresholds in $\mathscr{K}$ (X-axis). The analysis is provided for different graph explainers, and for the best *attribute* per *concept*.

7 Weakly Supervised Learning for Joint Whole-Slide Segmentation and Classification in Prostate Cancer

The ideas, methods and results presented in this chapter are published in:

- "Learning Whole-Slide Segmentation from Inexact and Incomplete Labels using Tissue Graphs", Valentin Anklin*, Pushpak Pati*, **Guillaume Jaume***, Behzad Bozorgtabar, Antonio Foncubierta-Rodríguez, Jean-Philippe Thiran, Mathilde Sibony, Maria Gabrani, Orcun Goksel. In *Medical Image Computing and Computer Assisted Interventions (MICCAI)*, 2021 (Anklin et al., 2021).
- "Weakly Supervised Learning for Joint Whole-Slide Segmentation and Classification in Prostate Cancer", **Guillaume Jaume***, Pushpak Pati*, Behzad Bozorgtabar, Jean-Philippe Thiran, Orcun Goksel, Maria Gabrani. In *Preprint*, 2021 (Jaume et al., 2021c).

GJ (the author of this thesis) is sharing first co-authorship with VA and PP on the first publication, and with PP on the second one. The ideas and concepts were conceived by GJ and PP and executed by GJ, PP and VA. The experiments were designed by GJ, VA and PP. MS validated the clinical soundness of the work. JP, OF, BB, AF, MG supervised and supported GJ in organizing his research. Both manuscripts were written by GJ and PP and subsequently revised by BB and OG.

7.1 Introduction

With the advancements in CompPath, several supervised CAD tools have been proposed proposed to assist pathology diagnosis across various tissue types and histopathology applications, *e.g.*, nuclei segmentation (Graham et al., 2019a; Verma et al., 2021), gland segmentation (Sirinukunwattana et al., 2017; Binder et al., 2019), tumor region detection (Bejnordi et al., 2019; Aresta et al., 2019), etc. Although these DL-based tools achieve remarkable performance, they often require task- and tissue-specific pixel or patch annotations on large datasets. Acquiring such annotations is laborious, time-consuming, and often infeasible.

To alleviate the burden of annotation requirements, weakly-supervised methods are proposed

that can leverage readily available WSI-level annotations. Most of these weakly-supervised methods, that are scalable to WSIs, focus on classification tasks, *e.g.*, MIL (Tellez et al., 2019a; Shaban et al., 2020) or compression-based representation learning (Tellez et al., 2019a; Shaban et al., 2020). Though methods classifying WSIs are important, their applicability is limited due to their poor ability to assist pathologists' focus during diagnosis (Wang et al., 2019b). To address this limitation with classification methods, semantic segmentation methods are desired that can delineate diagnostically relevant regions in WSIs and save pathologists' diagnosis time by directly guiding their focus to informative regions. A segmentation can enable the *quantification* of tumor regions for better patient stratification and tailored treatment selection. Further, complementing a WSI classification method by a pixel-level segmentation can ascertain the relevance of the WSI-level classification, thereby strengthening trust between the DL methods and pathologists. However, semantic segmentation of WSIs is more annotation-demanding, *i.e.*, requiring pixel-level labeling, compared to WSI classification. Therefore, weakly-supervised semantic segmentation (WSS) methods are imperative in histopathology diagnosis.

While WSS methods have shown great successes on natural images, they encounter several challenges when applied to histopathology images (Chan et al., 2021), as histopathology images (i) contain finer-grained objects with large intra-class variations (Xie et al., 2019); (ii) often include ambiguous boundaries among different histology components (Xu et al., 2017b); (iii) can be as large as several giga-pixels with arbitrary tissue shapes, such as WSIs. For instance, the methods by Xu et al. (2014); Hou et al. (2016); Jia et al. (2017); Xu et al. (2019a); Ho et al. (2021) perform WSS at patch-level. These methods are limited by their requirement of patch-level labels and their inability to incorporate global tissue microenvironment information for performing contextualized WSI segmentation. While Chan et al. (2019); Silva-Rodríguez et al. (2021) propose to analyse larger image-tiles compared to patches, they are constrained in terms of computational complexity and memory requirements to operate on WSIs in an end-to-end manner. The WSS method by Chan et al. (2019) requires *exact* fine-grained tile-level annotations, *i.e.*, a precise denomination of the presence of each lesion type in an image-tile during model training, which requires pathologists to annotate images beyond standard clinical needs and norms. On a different note, recent WSI classification methods propose to use learned attention weights or feature attribution techniques to highlight salient regions in a WSI that drive the model's prediction (Lu et al., 2021b; Tellez et al., 2019a). The identified salient regions are informative for visual assessment, but are insufficient, incomplete, and blurry for accurately delineating diagnostically relevant regions. Further, the saliency of a region signifies its relevance towards the model prediction, but do not always convey the class label of the region. In addition, these methods typically require densely overlapping patch-level predictions to obtain a granular saliency map, which is computationally expensive while working with WSIs.

In addition to the above shortcomings, the aforementioned approaches do not include uncertainty estimate analyses, which are crucial to understand *when* to trust the model predictions. Indeed, DL methods typically tend to produce overconfident predictions and do not indicate

when they are likely to be incorrect (Fort et al., 2019), especially when generalizing predictions to unseen cohorts. This can be partially explained by the lack of confidence and uncertainty estimates in neural network parameters, also known as *epistemic uncertainty*. Intuitively, epistemic uncertainty can be correlated to the inter-observer variability in pathology diagnosis, which is known to be high for challenging tasks. Each pathologist, with his/her experience, develops an own understanding of the task. Thus, pathologists can be considered as different "models", with different decision boundaries that induce uncertainty in challenging cases. Further uncertainty can be induced due to data, also known as *aleatoric uncertainty*. In pathology, aleatoric uncertainty is caused by, the difficulty of matching the continuum of histologic features to the diagnostic spectrum, intra- and inter-patient tumor heterogeneity, and visualization artifacts that create ambiguous cases. Consequently, *aleatoric* and *epistemic* uncertainty are inherently part of pathology practice and should be considered when developing CAD tools.

Given the above, it is imperative to develop a WSS method that can (i) operate on arbitrary and large histopathology images, *e.g.*, on WSIs; (ii) utilize both local and global context to conduct precise segmentation; (iii) perform simultaneous classification and segmentation; (iv) leverage readily available annotations in a clinical setting, without any task-specific assumptions or post-processing; and (v) provide reliable uncertainty estimates as confidence to diagnostic predictions as well as to detect any domain shifts when applied to new datasets.

To address the aforementioned requirements, we propose WHOLESIGHT, "Whole-slide SegmentatIon using Graphs for HisTopathology". Formally, WHOLESIGHT represents a histopathology image using a superpixel-based tissue-graph (TG), and transforms the segmentation task into a *node-classification* task. WHOLESIGHT incorporates both local and global inter-tissue-region relationships to perform contextualized segmentation. To account for *epistemic* uncertainty, we further propose two Bayesian variants of WSS based on MC-dropout (Gal and Ghahramani, 2016; Kendall and Yarin, 2017) (MCD) and deep ensembles (Lakshminarayanan et al., 2017; Fort et al., 2019) (DE), respectively. Our major contributions are:

- WHOLESIGHT, a novel weakly-supervised semantic segmentation and classification method that can scale to WSIs. WHOLESIGHT directly predicts the Gleason pattern associated to each pixel, *i.e.*, Benign (B), grade 3 (G3), grade 4 (G4) and grade 5 (G5), along with the WSI-level grade defined as the combination of the most common (*primary*, P) and the second most common (*secondary*, S) cancer growth patterns found in the image.

- A thorough evaluation of WHOLESIGHT on three prostate cancer datasets for Gleason pattern segmentation and Gleason grading, and comparison against state-of-the-art WSI classification algorithms.

- A study of the generalizability of WHOLESIGHT, WHOLESIGHT-MCD and WHOLESIGHT-DE when tested on *in-domain* and *out-of-domain* cohorts, including segmentation and

classification performance, uncertainty estimations and neural network calibration analyses.

7.2 Related work

7.2.1 Weakly-supervised histopathology image classification

Most of the weakly-supervised methods in CompPath are proposed to *classify* histopathology images, *i.e.*, tissue microarrays and whole-slides. EM-CNN is introduced in Hou et al. (2016), a patch-based method that is trained using image-level labels. It employs an Expectation Maximization (EM)-based method to identify discriminative patches by utilizing the inter-patch spatial relationships, and subsequently uses a decision fusion model to aggregate the patch-level predictions. A two-step approach is proposed in Campanella et al. (2019), which first identifies informative patches using a patch-level MIL framework, and then adopts a RNN-based strategy to aggregate patch-level predictions for WSI classification. Another MIL approach, CLAM, is proposed in Lu et al. (2021b) that learns class-level attention weights to discriminate diagnostically relevant regions. CLAM is further optimized by learning an instance-level clustering over the patches to constrain and refine the learned feature space. Differently, two-step compression-based procedures are proposed in Tellez et al. (2019a) and Shaban et al. (2020) to analyse WSIs. First, they extract patch-level embeddings using a network pre-trained on an auxiliary task (Tellez et al., 2019a; Shaban et al., 2020), *e.g.*, contrastive learning, or using unsupervised learning (Tellez et al., 2019a), *e.g.*, a Variational Auto-Encoder (VAE). Then, they build a compressed feature cube representation of the input WSI, which is further processed by a CNN classifier. Despite the success of these weakly-supervised classification approaches, they cannot directly be extended for semantic segmentation.

7.2.2 Weakly-Supervised histopathology image segmentation

A few methods in literature have been proposed to perform WSS of histopathology images. DWS-MIL is proposed in Jia et al. (2017), which trains a binary-classifier to generate pixel-level predictions, and then produces an image-level prediction using a softmax function. The network is trained to optimize the image-level prediction, and thereby improving the pixel-level predictions. A MIL-based label enrichment method, CAMEL, is proposed in Xu et al. (2019a) for WSS. It splits an image into latticed instances and automatically generates instance-level labels. After label enrichment, the instance-level labels are further assigned to the corresponding pixels, producing the approximate pixel-level labels and making fully supervised training of segmentation models possible. A deep multi-magnification network is introduced in Ho et al. (2021) which performs patch-wise multi-class tissue segmentation by using concentric patches across multiple magnifications. This method leverages scribble annotations of regions in WSIs during the training phase. HistoSegNet, proposed in Chan et al. (2019), performs WSS of histological tissue types in two steps. First, a CNN is trained

at tile-level using tile-level annotations to predict the presence of different tissue types in a tile. Then, GRAD-CAM, a feature attribution technique is employed to derive pixel-level class predictions. To further improve the segmentation, HistoSegNet employs a complex hand-crafted class-specific post-processing steps. As a main limitation, the aforementioned methods cannot perform WSS on giga-pixel WSIs using only image-level labels, and cannot adapt to WSIs of different sizes. Comparatively, WeGleNet proposed in Silva-Rodríguez et al. (2020) is scalable to WSIs. WeGleNet includes a multi-class segmentation layer and a global-aggregation layer to perform image-level classification during training and pixel-level prediction during inference. It aggregates class-wise pixel-level softmax activations to perform image-level task, and significantly upsample the pixel-level activations to segment an image. However, the method is insufficient to precisely delineate different lesions in an image, and is incomplete to highlight multiple occurrings of lesions. Further, it also requires to extract densely-overlapping patches to render fined-grained segmentation. In contrast, our proposed WSS approach can perform WSS by leveraging image-level labels, while efficiently scaling to WSIs with arbitrary shape and size.

7.2.3 Domain shift, generalization, and uncertainty in computational pathology

Domain shift and generalization

Building models that are in the same time robust to domain shifts and able to provide reliable uncertainty estimates is fundamental to deploy CAD tools in the real-world (Tellez et al., 2018, 2019b). Domain shifts are known to be challenging to model and detect in DL. This is prevalent in CompPath, where domain-level, *e.g.*, hospital-level, biases are introduced due to a variety of reasons, such as different staining protocols, manufacturing devices, materials, and scanning devices with respective color response (Aubreville et al., 2021). Nevertheless, several approaches have been proposed to reduce such domain shifts by developing data- and model-level adaptation mechanisms.

Stain normalization (Reinhard et al., 2001; Macenko et al., 2009; Vahadane et al., 2016; Ren et al., 2019) is a widely employed technique that reduces appearance variability across samples by using a reference image. It directly operates at data-level by standardizing the input in a reproducible way. Stain normalization is model-agnostic and has been shown to improve generalization performance of DL models (Tellez et al., 2018, 2019b). Differently, color augmentations are proposed to model staining variations, *e.g.*, by adding additive and multiplicative noise to the input (Tellez et al., 2018; Faryna et al., 2021). These techniques offer good compromise between ease of integration in DL pipelines and performance gain.

In another scenario, when (unlabeled) samples from the target domain are available in the training phase, domain adversarial training (Ganin et al., 2016; Aubreville et al., 2020) have been proven effective in domain adaptation. However, the availability of target domain samples for training is often impractical due to the lack of knowledge about where the model will be used, and limitations related to data privacy and regulations. Further, a pre-trained

model on a source domain can be fine-tuned by leveraging a few labeled target domain samples, but at the cost of compromising the generalization capabilities of the model.

Uncertainty estimation

While the aforementioned approaches propose various mechanisms to alleviate the impact of the distribution shifts, they do not address the scenario where the distribution on unseen cohorts is drastically changing. In this case, accurate *uncertainty* estimates are crucial to know *when* to trust the model – a task known to be challenging for neural networks that often provide over-confident predictions, as studied in (Guo et al., 2019; Lakshminarayanan et al., 2017; Fort et al., 2019). This may hinder real-life deployment in clinics, where CAD must be transparent.

However, CompPath research in uncertainty is scarce and remains an unexplored direction. Thagaard et al. (2020) benchmarked the detection of adenocarcinoma in H&E lymph node sections from breast cancer under various real-life distribution shifts. Their work concluded that Bayesian neural networks based on deep ensembles (Fort et al., 2019) and MC-dropout (Gal and Ghahramani, 2016; Kendall and Yarin, 2017; Fort et al., 2019) provide better uncertainty estimates than classical approaches. Our proposed generalization and uncertainty analysis further ascertains the findings of Thagaard et al. (2020) for WSI-level Gleason grading.

7.3 Methods

This section presents the proposed WHOLESIGHT methodology for scalable WSS of histopathology images. First, an input image is transformed into a tissue-graph (TG) representation, where the nodes and edges of the graph denote tissue regions and their interactions, respectively. Then, a GNN learns node-level embeddings that contextually characterize the tissue regions. The resulting node embeddings are processed by a *graph-classification head* for primary and secondary Gleason classification. At intermediate epochs during the training phase, a feature attribution technique and a node selection strategy are employed to determine pseudo labels for a subset of the nodes, which are further used to train a *node-classification head*. The outcomes of the *node-head* is used to segment the Gleason patterns in the image. An overview of WHOLESIGHT is provided in Figure 7.1.

7.3.1 Notation and preliminaries

Following the notation introduced in Chapter 1, we define an attributed graph $G \in \mathcal{G}$ as a triplet (V_G, E_G, H), where V_G and E_G represent the set of nodes and edges, $H \in \mathbb{R}^{|V| \times d}$ are node attributes, and $\mathcal{G}$ represents the set of graphs.

GNNs (Kipf and Welling, 2017; Xu et al., 2019b; Hamilton et al., 2017; Velickovic et al., 2018) are a class of neural architectures that can learn from graph-structured data. In a typical

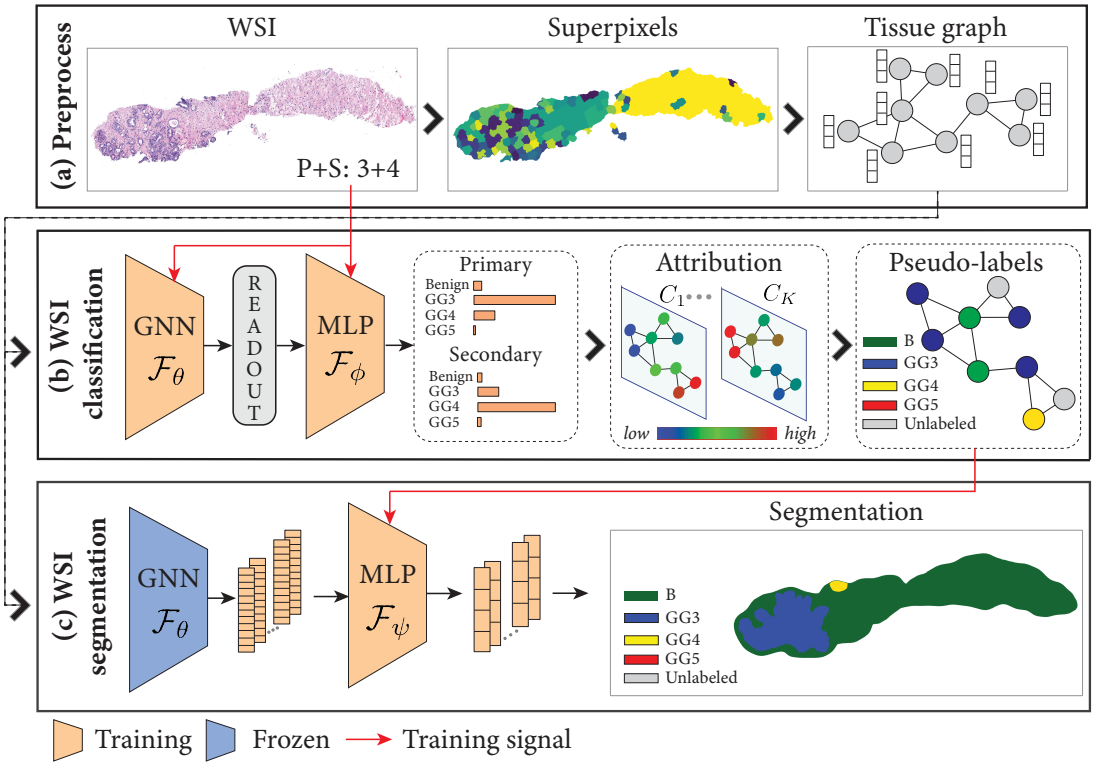


Figure 7.1 – Overview of the proposed WHOLESIGHT method. (a) In the preprocessing step, superpixels are detected to divide the input WSI into morphologically consistent tissue regions. Each tissue region is passed into a feature extractor to derive instance-level embeddings. The tissue regions and respective embeddings define the nodes and node features, respectively, of the TG. Adjacent tissue regions are further connected to each other to define the TG topology. (b) *Graph-classification head* to classify the TG representation of the WSI. The TG is passed to a GNN $\mathcal{F}_\theta$, followed by a readout, and MLP classifier $\mathcal{F}_\phi$ to predict the corresponding primary and secondary Gleason pattern. In a post-hoc step, a feature attribution method, followed by an importance-based node selection strategy derives node-level pseudo-labels. (c) *Node-classification head* to segment the WSI. The GNN $\mathcal{F}_\theta$ is re-used to obtain contextualized node-level embeddings. Afterwards, the pseudo-labels, derived from the graph-head, are used to train a node-level MLP classifier $\mathcal{F}_\psi$. The segmentation output is trivially obtained by mapping the node predictions to the input WSI.

message-passing GNN, the node features are iteratively updated via a two-step procedure to contextualize their feature representation in accordance with their neighborhood node information. In this work, we use a version of the GIN architecture Xu et al. (2019b), where the AGGREGATE step is based on a *mean*-operator, and the UPDATE step combines the *aggregated* features with the current node features $h(v)$ via a MLP. Formally, the AGGREGATE and the UPDATE steps are given as,

$$h^{(t+1)}(v) = \text{MLP}\Big(h^{(t)}(v) + \frac{1}{|\mathcal{N}(v)|}\sum_{u\in\mathcal{N}(v)} h^{(t)}(u)\Big) \tag{7.1}$$

The GNN is denoted as $\mathscr{F}_\theta$ that maps the graph nodes to embeddings, where θ are learnable parameters. For a graph classification, a fix-sized graph-level embedding h_G is derived by pooling the node-level feature representations $h^T(v),\ \forall v \in V_G$ by a READOUT step, *e.g.*, a mean-READOUT operation. Subsequently, the graph-level embeddings can be mapped to target classes by a neural network classifier $\mathscr{F}_\phi$, where ϕ are learnable parameters. Similarly, for a node classification task, the node-level feature representations $h^T(v),\ \forall v \in V_G$ can be classified by a neural network classifier $\mathscr{F}_\psi$, where ψ are learnable parameters.

Formally, classification aims to predict a target label $y \in \mathscr{K}$ for an input $x \in \mathscr{X}$, where $\mathscr{K}$ and $\mathscr{X}$ denote the set of classes and the set of inputs, respectively. Given a set of sample pairs $\{(x_i, y_i)\}_{i=1}^N$, where N is the number of samples and $(x_i, y_i) \sim p(x, y)$, the data likelihood can be expressed as $p(Y|X,\theta,\phi) = \Pi_{i=1}^N p(y_i|x_i,\theta,\phi)$. The optimal parameters $(\hat{\theta},\hat{\phi})$ are obtained by Maximum Likelihood Estimation (MLE), or equivalently by minimizing the Negative Log-Likelihood (NLL) $-\sum_{i=1}^N \log p(y_i|x_i,\theta,\phi)$. In practice, NLL is expressed as a cross-entropy loss, where the model weights are updated by Stochastic Gradient Descent (SGD), or a similar gradient-based optimizer. In a graph classification setting, a sample pair is denoted as $(y_G, G),\ y_G \in \mathscr{K}_\mathscr{G},\ G \in \mathscr{G}$. In node classification, a sample pair is denoted as $(y_V, v),\ y_V \in \mathscr{K}_V,\ v \in \mathscr{V}$. For the considered task at hand, the set of graph- and node-level classes are the same, simplifying notation to $\mathscr{K} := \mathscr{K}_\mathscr{G} = \mathscr{K}_V$.

We further introduce the notion of model *calibration*. Intuitively, the probability of outcomes, *i.e.*, confidence scores, of a calibrated model should match its performance. For example, the samples predicted with an average confidence of 60% by a model should have an average accuracy of 60%. Formally, for a given network, $f : \mathscr{X} \to \mathscr{K}$, and $p(X, Y)$ a joint distribution over the data and the labels, $f(x)$ is said to be calibrated with respect to p if, $\mathbb{E}_p[Y|f(X) = \beta] = \beta$, $\forall \beta \in [0, 1]$. The *calibration* can be visualized with a *reliability diagram* (DeGroot et al., 1983). Namely, all the samples in the dataset are assigned to bins according to their predicted confidence scores by the network. Then, the network performance, *e.g.*, accuracy, is computed for all the samples in each bin. The network performance is plotted against the binned confidence scores, where deviations from the diagonal represent uncalibrated bins.

7.3.2 Preprocessing and tissue-graph construction

The input H&E stained images in the dataset are first stain-normalized using the algorithm proposed by Vahadane et al. (2016) to reduce any appearance variability across the images due to tissue preparation, such as different specimen preparation techniques, staining protocols, fixation characteristics, and imaging device characteristics. In the next step, a stain normalized image is transformed into a TG (Figure 7.1(a)), where the nodes and the edges of the TG denote tissue regions and inter-tissue interactions, respectively. Motivated by Bejnordi et al. (2015), we consider superpixels as the visual primitives to encode the tissue regions for this work. In comparison to rectangular patches, superpixels are flexible units to accommodate arbitrary shapes in accordance with the local homogeneity of the tissue in an image. The homogeneity

constraint also restricts the superpixels to span across multiple distinct structures and include different morphological regions.

We briefly remind the key steps for constructing a TG: (i) the construction of superpixels to define the nodes V_G, (ii) characterization of the superpixels to define the node features H, and (iii) the construction of the graph topology to define the edges E_G. For identifying the superpixels in an input image, a two-step procedure is adopted. First, unsupervised SLIC algorithm (Achanta et al., 2012) emphasizing on space proximity is employed on the image to produce over-segmented superpixels. The SLIC algorithm is applied on a low magnification of the image to capture homogeneity, while offering a good compromise between granularity and smoothing-out noise. In the second step, the over-segmented superpixels are hierarchically merged according to their channel-wise color similarity at high magnification. The color similarity is quantified in terms of channel-wise 8-bin color histograms, mean, standard-deviation, median, energy, and skewness. The resulting merged tissue regions form the nodes of the TG. The merging allows to semantically group the superpixels and render meaningful tissue regions. In addition, the merging reduces the node complexity of the TG, thereby enabling the scaling of TG to large dimensional histopathology images and contextualization to distant nodes.

To characterize the nodes of the TG, we extract morphological and spatial features from the tissue regions constituting the nodes. Considering the arbitrary dimensions of the superpixels, a two-step process is adopted to extract deep learning-based morphological features. First, patches of size 144×144 pixels are extracted from a superpixel, resized to 224×224 size, and encoded into 1280-dimensional features by processing through a MobileNetV2 network (Sandler et al., 2018) pre-trained on ImageNet (Deng et al., 2009a). Then, the corresponding node-level morphological features are computed as the mean of the individual patch-level features. Further, spatial features of the nodes are computed by normalizing the superpixel centroids by the image dimensions. The normalization ensures the invariability of the spatial features with respect to the varying dimensions of the input histopathology images. Finally, the TG topology is defined by constructing a RAG (Potjer, 1996) using the spatial connectivity of superpixels. To this end, we assume that adjacent tissue regions biologically interact the most, and thus should be connect in the TG topology.

7.3.3 Contextualized node embeddings

Given a TG, we aim to learn discriminative node embeddings (see Figure 7.1(b)) by utilizing the context information of the nodes, *i.e.*, the tissue micro-environment and the inter-tissue interactions. The contextualized node embeddings are subsequently used for WSI classification and WSS. To contextualize the node embeddings, we use a GIN (Xu et al., 2019b) graph neural network, denoted as $\mathscr{F}_\theta$ and parametrized by the learnable parameters θ. Since GNNs can operate on graphs of arbitrary and varying sizes, they allow to encode histopathology images represented in the form of TGs without the need for tile-based processing. As the

discriminative information, dependent on the sub-graph structures, can lie at different abstraction levels in the GNN, we employ a Jumping Knowledge (JK) strategy to incorporate multi-level node representations. Namely, the final node-level embedding after T GIN-layers is defined as,

$$h^{(T)}(v) = \text{CONCAT}(h^{(t)}(v),\ \forall t \in \{1, ..., T\}) \tag{7.2}$$

where, CONCAT denotes a concatenation operation.

7.3.4 WSI classification

Following the contextualized node embeddings, a *graph-classification head* is employed to classify the TG by leveraging image-level *inexact* labels. To this end, first, a READOUT averages out the information from all the nodes $h^{(T)}(v),\ \forall v \in V_G$ to build a fix-sized graph-level embedding h_G. Subsequently, the graph-level embedding is fed to a multi-task classifier for primary and secondary Gleason grading. Specifically, the classifier is composed of two parallel MLPs, denoted as $F_\phi = \{F_{\phi_1}, F_{\phi_2}\}$, which are parametrized by trainable parameters $\phi = \{\phi_1, \phi_2\}$. The two MLPs individually predict the primary, *i.e.*, the worst Gleason pattern, and secondary, *i.e.*, the second worst Gleason pattern, in the WSI. Each MLP solves a multi-class problem with $|\mathscr{K}|$ Gleason pattern classes, *i.e.*, benign, grade 3, grade 4, and grade 5. The final Gleason grade is derived as the sum of the predicted primary and secondary Gleason patterns. $\mathscr{F}_\theta$ and $\mathscr{F}_\phi$ are optimized jointly by minimizing the weighted multi-label cross-entropy loss,

$$\mathscr{L}_G = \lambda \mathscr{L}_{CE}(y_{G_P}, \hat{y}_{G_P}) + (1-\lambda)\mathscr{L}_{CE}(y_{G_S}, \hat{y}_{G_S}) \tag{7.3}$$

where, P and S denote the primary and the secondary labels of ground truth y_G and prediction $\hat{y}_G$, and $\lambda \in [0, 1]$ is a hyper-parameter used to balance the two terms. Gleason grading is typically imbalanced, where WSIs with higher grade patterns are less frequent. To address this, we define class-weights as $w := \{\log(\frac{\sum_i N_i}{N_i}),\ i = \{1, ..., |\mathscr{K}|\}\}$, where N_i is the count of class-wise Gleason patterns. The weights are designed such that a higher value is assigned to classes with lower frequency.

7.3.5 Weakly supervised semantic segmentation

The nodes in a TG are identified by superpixels that denote morphologically homogeneous tissue regions. Since each Gleason pattern is characterized by *distinct* morphological patterns, we assume that each tissue region, depicted by a node of the TG, includes a *unique* Gleason pattern. Thereby, the WSI segmentation task is transformed into a classification task of the nodes in the TG. In the presence of only image-level labels, the node classification task is achieved in two steps. First, pseudo-node labels are generated by leveraging the image-level annotations, and subsequently the pseudo-node labels are used to train a node classifier.

Pseudo node label generation

Following the image-level classification in Section 7.3.4, a post-hoc *feature attribution* technique is employed to measure the importance of each node for the TG classification. Specifically, we use GRAPHGRAD-CAM (Pope et al., 2019; Jaume et al., 2021b), an extension of GRAD-CAM (Selvaraju et al., 2017) technique to operate with GNNs. For a graph G, GRAPHGRAD-CAM produces class-wise node attribution maps, A_k, $\forall k \in \mathscr{K}$. The attribution maps highlight the importance $\forall v \in V_G$ towards the classification of G into $|\mathscr{K}|$ categories, as demonstrated in Figure 7.1. Given the importance scores of a node $v \in V_G$ towards $|\mathscr{K}|$ classes, a simple and straightforward approach is to assume that the class label of v is $k \in \mathscr{K}$ if the highest importance score corresponds to class k. At this stage, an *argmax* operation across the class-wise importance scores $\forall v \in V_G$ can be considered to classify the nodes. However, such node classification strategy carries several disadvantages.

- An *argmax* operation for a node greedily selects the class label with the highest importance score. However, some nodes only marginally contribute to the graph classification, *e.g.*, background nodes, and bear low importance scores for all $k \in \mathscr{K}$. An *argmax* operation would confidently label such nodes into one of the $\mathscr{K}$ classes, which reduces confidence in the node classification.

- The class labels of the nodes, that highly contribute towards a certain class, cannot not be guaranteed to be the same as the corresponding class label. Formally, if the set of nodes $V_k \subset V$ have high importance scores for class k, then the class labels of V_k are not ensured to be k, *e.g.*, a node $v \in V_k$ can be an evidence of the *absence* of all classes $\mathscr{K} \setminus \{k\}$, thus bearing high importance for classifying the graph as k, while not being of this class.

- GRAPHGRAD-CAM does not necessarily highlight all the nodes that belong to a class in the corresponding class attribution map. Depending on the complexity of a classification task, a classifier may utilize only a subset of the informative nodes corresponding to a class to predict the label of the graph. Formally, if the set of nodes $V_k \subset V$ have high importance scores for class k, then V_k may not include all the nodes in $\mathscr{V}_k \subset V$ that have the actual label k, *i.e.*, $V_k \subset \mathscr{V}_k$.

- There are several feature attribution techniques in literature that can be employed to assign node-wise importance scores and perform node classification. However, as demonstrated in Jaume et al. (2021b), differences in the underlying mechanisms of these techniques lead to different node-wise importance scores. Therefore, a single feature attribution technique, *e.g.*, GRAPHGRAD-CAM, may not be trusted for a score-based node classification.

Therefore, we devise a strategy to use the highlighted nodes by GRAPHGRAD-CAM as pseudo-labels to train a *node-classification head*. The strategy aims to create pseudo-labels while

minimizing the class-wise false positives and false negatives. Specifically, for a graph G with Gleason score $P+S$, such that $P,S \in \mathscr{K}$, we compute the node importance scores I_P and I_S $\forall v \in V_G$. I_P and I_S are computed by using un-normalized GRAPHGRAD-CAM on the P-th class and the S-th class in the primary and secondary graph-classification heads. Since the importance scores by GRAPHGRAD-CAM are unbounded, employing a fixed threshold on the importance scores across all samples is sub-optimal. Therefore, we select the top n% nodes, denotes as V_P and V_S, based on the respective importance scores I_P and I_S. n is a hyperparameter, which is tuned on during the training phase. For a node $v \in V_P$ and $v \in V_S$, we compute the argmax$(I_P(v), I_S(v))$ to assign v into either of the sets. This ensures that $V_P \cap V_S = \emptyset$. Subsequently, we label the nodes $v \in V_P$ as P and the nodes $u \in V_S$ as S. This process ensures to select the most important set of nodes corresponding to the ground truth image-level label of G, and create the pseudo-labels, denoted as $y_{\tilde{V}}$. Continuing this process for all the TGs in the dataset produces pseudo-node labels across all classes, denoted as $Y_{\tilde{V}}$.

Node classification

The pseudo-node labels $Y_{\tilde{V}}$ are used to train a *node-classification head*, as shown in Figure 7.1. Specifically for a graph G, we extract the node embeddings $h^{(T)}(v)$, $\forall v \in V_G$ using $\mathscr{F}_{\hat{\theta}}$, where $\hat{\theta}$ are the parameters from the graph classification in Section 7.3.4. $\mathscr{F}_{\hat{\theta}}$ is kept frozen during the node classification to ensure that the *same* GNN backbone can be used for both segmentation and classification, thereby reducing the number of trainable parameters. The node embeddings are processed by an MLP classifier $\mathscr{F}_{\psi}$, parameterized by learnable parameters ψ, to predict the pseudo-node labels. The *node-classification head* $\mathscr{F}_{\psi}$ is trained by minimizing a weighted multi-class cross-entropy objective. Similar to the graph classification setting, class-weights are defined as $w := \{\log(\frac{\sum_i N_i}{N_i}),\ i = \{1,...,|\mathscr{K}|\}\}$, where N_i is the number of annotated nodes of class i. The node-wise predicted class labels are finally used to obtain the segmentation prediction.

We refer to our proposed method, the simultaneous WSI classification and pseudo-node labeling-based WSS, as WHOLESIGHT. Noticeably, unlike Chan et al. (2019), WHOLESIGHT does not involve any customized post-processing, thus being a generic method that can be applied to various organs, tissue types, segmentation tasks, etc.

7.3.6 Extension to Bayesian models

We propose two Bayesian variants of WHOLESIGHT to incorporate uncertainty estimates into model predictions. We assume that *aleatoric* uncertainty, *i.e.*, data uncertainty, is already modeled during network training and reflected in the predicted probabilities of WHOLESIGHT. Since *epistemic* uncertainty is not explicitly captured by WHOLESIGHT, we propose to model it using WHOLESIGHT-MCD based on MC-dropout (Gal and Ghahramani, 2016; Kendall and Yarin, 2017) as well as WHOLESIGHT-DE based on deep ensembles (Lakshminarayanan et al., 2017; Fort et al., 2019). These methods are built on the fact that there exist several sets of

parameters that can explain a given dataset equally well, *i.e.*, a set of WSIs and WSI labels. The underlying principle of these methods aims to utilize multiple optimal models to capture the variations in the decision boundaries of the individual models, thereby accounting for the epistemic uncertainty. These methods are also crucial when generalizing to unseen cohorts, including distribution shifts in the data.

Deep Ensembles

Deep ensembles are realized by training several models with *different network initializations*, herein exploring diverse modes in function space. In our case of graph classification, recall that the conditional distribution $p(y_G|G,\theta,\phi)$ is approximated by our proposed network $\mathscr{F}_\phi(\mathscr{F}_\theta(G))$, which learns an optimal set of parameters $(\hat{\theta},\hat{\phi})$ with MLE. Using different network weight initializations, we can learn *different* optimal parameters $\{\hat{\theta}^{(m)},\hat{\phi}^{(m)}\}_{m=1}^{M}$, where $m \in \{1,...,M\}$ refers to different models. Then, for a test sample $G^* \in \mathscr{G}$, WHOLESIGHT-DE output is obtained by computing the average prediction from all the models, *i.e.*,

$$\hat{p}(y_G^*|G^*) := \frac{1}{M}\sum_{m=1}^{M} p(y_G^*|G^*,\hat{\theta}^{(m)},\hat{\phi}^{(m)}) \tag{7.4}$$

For node classification and WSI segmentation, a similar approach is employed where $p(y_V|v,\theta,\psi)$ is approximated by $\mathscr{F}_\psi(\mathscr{F}_\theta(v))$.

MC-dropout

MC-dropout (Gal and Ghahramani, 2016; Gustafsson et al., 2019) follows the same principle to propose a modification of the use of *dropout* layer in the network. Unlike the standard DL networks which utilize dropout only during training, MC-dropout proposes to retain the dropout layers during inference as well. Owing to the dropout layer that randomly switches off some neurons in the network, during inference, each forward pass operates on a different network defined as a *random* subset of the original network. The randomly sampled networks can be viewed as an *ensemble* of networks that provide different decision boundaries and thereby different predictions. As in deep ensembles, the output WHOLESIGHT-MCD predictions are obtained by averaging the network predictions over N passes with different dropout patterns.

7.4 Experiments

7.4.1 Datasets

We evaluate our proposed method on three prostate cancer datasets acquired from three independent data sources, consisting of whole-slide prostate cancer needle biopsies. We use these datasets for simultaneously segmenting Gleason patterns in the WSIs and classify the WSIs into different Gleason grades. The Gleason patterns range from grade 3 (G3), characterized by

moderately differentiated nuclei and the presence of poorly-formed and cribriform glands, to grade 4 (G4), that include poorly differentiated nuclei and irregular masses, to grade 5 (G5), characterized by even less differentiated nuclei and lack or only occasional glands. Normal glands and non-epithelial tissue regions are categorized as benign (B). The Gleason grade is estimated from a Gleason score which is presented as *primary* + *secondary*, where the *primary* and the *secondary* denote the worst and the second worst Gleason patterns, respectively. Details of the datasets are presented as follows:

Radboud dataset

The Radboud dataset (Bulten et al., 2020) is composed of 5,759 core needle biopsies extracted from 1,243 patients. The data were acquired between January 1, 2012, and December 31, 2017, from patients who underwent prostate biopsy for suspected cancer at the Radboud University Medical Center. All the slides were scanned with a 3D Histech Panoramic Flash II 250 scanner at 20× magnification (pixel resolution 0.24μm), and were further downsampled to 10×. The annotations include WSI-level Gleason grade extracted from patient records. Further, noisy pixel-level segmentation masks of Gleason patterns on the WSIs were made available as part of the Prostate cANcer graDe Assessment (PANDA) challenge. These segmentation masks were cleaned for the purpose of Gleason pattern segmentation by using standard image manipulation techniques, such as contextualized noise removal, hole filling, and edge smoothing. In the absence of large public datasets that consist of pixel-level annotated prostate cancer WSIs, we utilized the Radboud dataset for the development and evaluation of our methods.

Karolinska dataset

The Karolinska dataset (Ström et al., 2019) comprises 5,662 core needle biopsies extracted from 1,222 patients. The data were acquired on men aged between 50 and 69 years, between 2012 and 2015 in various hospitals in Stockholm, Sweden. The slides were scanned with a Hamamatsu C9600-12 and an Aperio Scan Scope AT2 scanner at 20× magnification, with pixel resolution of 0.45202μm and 0.5032μm, respectively. All the biopsies were annotated by an expert uro-pathologist for Gleason grading.

Sicap dataset

The Sicap dataset (Silva-Rodríguez et al., 2020) contains 18,783 patches of size 512×512 with *complete* pixel-level annotations and WSI-level Gleason grades from 155 WSIs extracted on 95 patients. As the original dataset is composed of patches, the original WSIs and annotation masks were reconstructed by stitching the patches. The WSIs were scanned at 40× resolution with a Ventana iS-can Coreo scanner, and further downsampled to 10× magnification for processing. Pixel- and WSI-level annotations were acquired by a group of expert urogenital

pathologists at the Hospital Clínico of Valencia.

Each dataset is split into train, validation, and test in a ratio of 60%, 20%, and 20% at Gleason grade-level, using a random stratified partition that preserves the percentage of samples in each class. No further sample-level analysis was performed to partition the data. The Gleason grade-wise dataset distribution is displayed in Figure 7.2, which highlights the different class-level imbalances across the three datasets. Karolinska dataset is more skewed towards benign and low-grade Gleason categories. The Gleason grade-wise distribution is the most balanced in the Radboud dataset. Notably, all three datasets contain a lower fraction of high-grade Gleason categories.

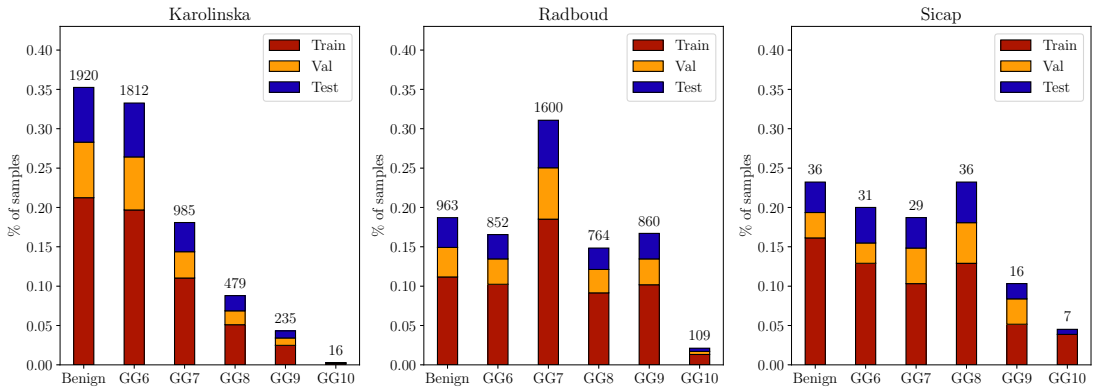


Figure 7.2 – Class distribution of the Karolinska, Radboud and Sicap datasets.

7.4.2 Implementation and metrics

We implemented our proposed method using PyTorch (Paszke et al., 2019), DGL (Wang et al., 2019a), and Histocartography (Jaume et al., 2021a). The experiments were conducted on NVIDIA Tesla P100 GPUs and POWER9 CPUs.

To develop the WHOLESIGHT network architecture, the GNN backbone $\mathscr{F}_\theta$, the *graph-classification head* $\mathscr{F}_\phi$, and the *node-classification head* $\mathscr{F}_\psi$ were developed by setting and optimizing their respective hyperparameters. First, $\mathscr{F}_\theta$ and $\mathscr{F}_\phi$ were trained by using image/graph-level labels, and afterwards pseudo-node labels were created to train $\mathscr{F}_\psi$. The segmentation output was obtained via node classification from $\mathscr{F}_\psi$. The number of GIN layers in $\mathscr{F}_\theta$ are optimized for the values $\{3,4,5\}$, where the UPDATE function was defined as a 2-layer MLP with 64 hidden units, and ReLU activations. The *graph-classification head* $\mathscr{F}_\phi$ contains two heads for classifying *primary* and *secondary* Gleason categories, where each head consists of a 2-layer MLP with 128 hidden units and ReLU activations. The *node-classification head* $\mathscr{F}_\psi$ contains a 2-layer MLP with 128 hidden units and ReLU activations.

For the Sicap dataset, which consists of a few WSIs, node-level augmentation techniques are employed to augment the graph dataset. Specifically, random node rotations $\{90,180,270\}$ degrees, and horizontal and vertical mirroring are used for augmenting the nodes. The batch

size and the learning rate were optimized from $\{4, 8, 16\}$ and $\{10^{-4}, 5 \times 10^{-4}, 10^{-3}\}$ set of values, respectively. Dropout layers with dropout rates 0.2, 0.5, and 0.5 were included in the MLPs belonging to $\mathscr{F}_\theta$, $\mathscr{F}_\phi$, and $\mathscr{F}_\psi$, respectively.

Following the hyperparameter tuning, eight WHOLESIGHT models were trained with *different* network initializations. The reported WHOLESIGHT results correspond to the mean and standard deviation obtained over these eight models. A similar approach was employed for WHOLESIGHT-MCD, where each model was run 25 times on different sampled networks created randomly by using the dropout layers. WHOLESIGHT-DE was defined by randomly sampling five out of the eight trained models. This process was repeated eight times to obtain different ensemble-based predictions. All the algorithms were trained with Adam optimizer (Kingma and Ba, 2015).

The model selection criteria during training relied on the version of the WHOLESIGHT method. For the first version, a model with the best Gleason grade weighted-F1 on the validation set was selected. In contrast, the model with the best node-classification weighted-F1 score on the validation set was selected for the other two versions. For creating the pseudo-node labels, several percentages of the most important nodes were selected, where the experimented percentage values were $\{5, 10, 15, 20\}$.

Classification metrics

WSI classification performance is measured by the weighted-F1 score of the Gleason grade between the ground truth and predicted labels. Additionally in accordance with the prior work (Bulten et al., 2020, 2021), we report the quadratic kappa score (κ^2) of the predicted ISUP grade (Epstein et al., 2005, 2014). ISUP grading is an alternative grading system whose correspondence with Gleason grading is defined as, Benign → ISUP-0, GG-(3+3) → ISUP-1, GG-(3+4) → ISUP-2, GG-(4+3) → ISUP-3, GG-8 → ISUP-4, and GG≥9 → ISUP-5. κ^2 incorporates for the level of disagreement between the prediction and ground truth labels. For example, for a sample with Gleason grade 6, predicting a grade 10 is penalized more compared to predicting a grade 7.

Segmentation metrics

The segmentation performance is measured by the Dice score between the ground truth and the predicted Gleason pattern segmentation masks. The Dice score is equivalent to F1-score at pixel-level predictions. Given the imbalance of the Gleason patterns in the datasets, we also report the per-pattern Dice score.

Uncertainty metrics

Following the previous work of Gomariz et al. (2021), we evaluate the classification and segmentation uncertainties by computing the Brier score s_{B} (lower is better) and the NLL s_{NLL} (lower is better) over a set of N unseen test samples, expressed as,

$$s_{\mathrm{B}} = \frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{|\mathscr{K}|}(y_i - \hat{y}_i)^2, \quad s_{\mathrm{NLL}} = -\frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{|\mathscr{K}|} p(y_i)\log\hat{p}(y_i) \tag{7.5}$$

Intuitively, the uncertainty estimates will be good when the model performance is high, and when the misclassified samples are not highly confident in their predictions.

Calibration metrics

Reliability diagrams provide an intuitive understanding of model calibration. To quantify the observations in a reliability diagram, we use the Expected Calibration Error (ECE) metric (Kumar et al., 2018). It computes the weighted average deviation of the confidence scores over all the bins. Formally, it is expressed as,

$$c_{\mathrm{ECE}} = \sum_{b=1}^{B}\frac{N_b}{N}|\mathrm{acc}(b) - \mathrm{conf}(b)|, \tag{7.6}$$

where n_b represents the number of samples in bin b, $\mathrm{acc}(b)$ and $\mathrm{conf}(b)$ denote the accuracy and average confidence of samples in the bin b, respectively.

7.4.3 Baselines

We compare our proposed WHOLESIGHT with state-of-the-art WSI classification and two versions of WHOLESIGHT. The variants of WHOLESIGHT are denoted as WHOLESIGHT(Graph, GRAPHGRAD-CAM) and WHOLESIGHT(Multiplex, NC), which are trained using only image-level supervision and multiplexed supervision (both image- and pixel-level labels), respectively.

WHOLESIGHT(Graph, GRAPHGRAD-CAM)

We propose this variant of WHOLESIGHT that uses only image/ graph-level supervision during training. Compared to the proposed WHOLESIGHT method, this baseline contains only the GNN backbone $\mathscr{F}_\theta$ and the graph-classification head $\mathscr{F}_\phi$. It does not create or utilize pseudo labels, and the segmentation output is obtained by taking the *argmax* over the class-wise GRAPHGRAD-CAM attribution maps.

WHOLESIGHT(Multiplex, NC)

We propose this variant of WHOLESIGHT that leverages both *inexact* image- and *complete* pixel-level supervision during training. It acts as the upper bound for WHOLESIGHT method. As pixel-level annotations are available, the node-classification head is trained using ground-truth node-level labels, instead of generated pseudo-node labels. It constitutes of the same GNN backbone $\mathscr{F}_\theta$, graph-classification head $\mathscr{F}_\phi$, and node-classification head $\mathscr{F}_\psi$ as the WHOLESIGHT architecture. In this setting, $\mathscr{F}_\theta$, $\mathscr{F}_\phi$, and $\mathscr{F}_\psi$ are trained jointly by optimizing a multi-task objective, *i.e.*, WSI-level primary and secondary Gleason score prediction along with node-level Gleason pattern prediction. This variant of WHOLESIGHT was proposed in our preliminary work, as described in Anklin et al. (2021).

CLustering-constrained Attention Multi Instance Learning (CLAM)

CLAM (Lu et al., 2021b) is a clustering-constrained attention MIL approach designed for WSI classification. Our experiments are based on the publicly available implementation of CLAM [1]. Minor modifications were performed to adapt the algorithm for a multi-task objective, *i.e.*, primary and secondary Gleason score classification. Specifically, patches of size 256×256 were extracted from a WSIs. Each patch was further processed by a ResNet50 model pretrained on ImageNet, where features after the third residual block were extracted with an adaptive mean-spatial pooling operation, which resulted in a 1024-dimensional feature representation. The attention module used a self-attention network with sigmoid activations and a 0.25 dropout. The clustering module that learns class-level representations was trained by using outputs of the attention network as pseudo-labels and a smooth top1 SVM loss. The attention-weighted patch features were finally passed to a linear classifier for classifying the primary and secondary Gleason scores.

Neural Image Compression (NIC)

NIC (Tellez et al., 2019a) creates feature cube representations of WSIs to learn a mapping between deep patch features and WSI-level class labels. Our implementation and experiments are partially based on the publicly available implementation [2], which required to be completed with training utilities, data loaders, and model translation in PyTorch. Specifically, input WSIs were resized to the dimensions of the largest WSI in our datasets with padding. It allowed associating each WSI to WSI-level label without further processing. Different patch feature extraction strategies were experimented to extract the compressed WSI representations. In our experiments, we found that NIC with BiGAN features (see Tellez et al. (2019a) for implementation details) led to the best performance. A custom CNN with eight convolutional layers was trained from scratch, where each layer has 128 channels, a batch normalization module, 0.2 dropout, and stride 1. As a significant portion of the input is background, the

[1] CLAM publicly available code: https://github.com/mahmoodlab/CLAM
[2] NIC publicly available code: https://github.com/davidtellez/neural-image-compression

average pooling was replaced by max-pooling to extract the most relevant regions per channel. Then, the primary and secondary Gleason pattern classifiers were implemented as 2-layer MLPs with 128 channels and LeakyReLU activations. The network was trained with a multi-class cross-entropy loss.

For all the baselines, a hyper-parameter search was conducted to find the best learning rate and batch size, if applicable. Subsequently, eight models were re-trained from scratch with the optimal set of parameters. For each experiment, we report the average and standard deviation over these runs without further model selection.

7.4.4 WSS performance analysis

Training setting

We study the classification and segmentation performance of the proposed WHOLESIGHT method, and compare against the aforementioned baselines on three datasets, *i.e.*, Karolinska, Radboud, and Sicap datasets. These evaluations measure the standalone applicability of the WHOLESIGHT method across the independent train and test datasets.

Results analysis

Table 7.1 presents the classification and segmentation results on the Sicap dataset. The analyses are performed under two supervision settings, namely *complete* ($\mathscr{C}$) and *inexact* ($\mathscr{IE}$). The $\mathscr{C}$ setting utilizes both *inexact* image-level labels and the pixel-level annotations. Whereas, the $\mathscr{IE}$ setting only uses the *inexact* image-level labels. WHOLESIGHT reaches 39.3% average Dice score, which significantly outperforms WHOLESIGHT(Graph, GRAPHGRAD-CAM) by +8.6% in absolute. Further, WHOLESIGHT significantly outperforms HistoSegNet in terms of both classification and segmentation metrics. WHOLESIGHT(Multiplex, NC), which acts as the upperbound, results in slight improvement in classification and a significant gain in segmentation compared to WHOLESIGHT. The per-class Dice scores indicate that the benign patterns, that constitute most of the tissue area, have a high detection rate compared to less occurring Gleason patterns. For the classification task, WHOLESIGHT outperforms NIC and CLAM methods both in terms of Gleason grade weighted-F1 and ISUP κ^2. However, considering the small size of the Sicap test set, the classification performance assessment on the Radoud and Karolinka datasets reveal a more confident picture.

Table 7.2 presents the classification and segmentation results on the Radboud dataset. WHOLESIGHT renders an absolute gain of +10.33% in average Dice score over WHOLESIGHT(Graph, GRAPHGRAD-CAM). This confirms the utility of pseudo-node labels for a superior segmentation. WHOLESIGHT(Multiplex, NC) remains a good upper-bound with an average Dice score of 64.99 ± 0.4. The observations of class-wise Dice scores are consistent with Sicap, where the benign patterns have a high detection rate, followed by G3, G4, and G5 patterns. As the Radboud dataset includes more G5 patterns than Sicap, we observe a significant gain in detecting high-grade patterns. For the

Annot.	Method	per-class Dice				avg. Dice	GG wF1	ISUP κ^2
		Benign	Grade3	Grade4	Grade5			
$\mathscr{G}$	WHOLESIGHT (Multiplex, NC)	91.1±1.0	39.4±1.6	52.9±1.4	10.6±5.4	48.7±1.3	55.0±1.7	86.2±3.1
$\mathscr{I}\mathscr{E}$	NIC(Tellez et al., 2019a)	-	-	-	-	-	35.3±5.0	44.5±14.2
	CLAM(Lu et al., 2021b)	-	-	-	-	-	53.8±3.5	61.8±5.5
	HistoSegNet (Silva-Rodríguez et al., 2020)	71.5±1.4	1.5±0.7	8.4±0.9	1.6±0.3	22.4±0.3	16.7±4.3	36.7±2.8
	WHOLESIGHT (Graph, GRAPHGRAD-CAM)	65.5±2.3	23.3±4.2	30.0±5.5	4.1±1.4	30.7±2.1	**54.1±4.1**	**79.2±2.9**
	WHOLESIGHT (Graph + Pseudo, NC)	**73.0±3.1**	**34.7±1.2**	**43.8±5.3**	**5.7±0.4**	**39.3±1.4**	**54.7±4.6**	**81.4±5.2**

Table 7.1 – Classification and segmentation results on Sicap dataset. The best performances for using image-level supervision are highlighted in **bold**.

classification task, the observations are consistent with the observations on the Sicap dataset. Noticeably, the complementarity of the image- and pixel-level annotations results in a better classification performance for WHOLESIGHT(Multiplex, NC) than WHOLESIGHT.

Annot.	Method	per-class Dice				avg. Dice	GG wF1	ISUP κ^2
		Benign	Grade3	Grade4	Grade5			
$\mathscr{G}$	WHOLESIGHT (Multiplex, NC)	91.6±0.1	64.3±0.3	65.9±0.8	38.2±1.1	65.0±0.2	61.7±0.4	76.3±1.3
$\mathscr{I}\mathscr{E}$	NIC(Tellez et al., 2019a)	-	-	-	-	-	35.1±1.2	45.0±2.2
	CLAM(Lu et al., 2021b)	-	-	-	-	-	55.8±1.1	73.7±1.7
	WHOLESIGHT (Graph, GRAPHGRAD-CAM)	63.8±2.3	23.8±3.8	22.6±1.9	12.1±0.7	30.6±1.0	**58.0±0.8**	**73.8±1.6**
	WHOLESIGHT (Graph + Pseudo, NC)	**83.8±0.6**	**36.3±1.1**	**23.1±2.3**	**20.6±0.3**	**40.9±0.5**	**58.0±0.8**	**73.8±1.6**

Table 7.2 – Classification and segmentation results on Radboud dataset. The best performances for using image-level supervision are highlighted in **bold**.

Table 7.3 presents the classification results on the Karolinska dataset. In the absence of ground truth pixel-level annotations, the segmentation performances could not be computed. WHOLESIGHT outperforms NIC and produces comparable classification performance with respect to CLAM. The Gleason grade weighted-F1 score is higher for the Karolinska dataset compared to Radboud. This is due to the presence of more high-grade Gleason grade WSI in the Karolinska dataset. This observation is substantiated by the confusion matrix of Gleason grade classification for the WHOLESIGHT-DE method, as shown in Figure 7.3.

7.4.5 Generalization: performance, uncertainty, and calibration

Training setting

To study the generalization capability of WHOLESIGHT, we propose a modified training setting. Specifically, we build a new training dataset that comprises Karolinska and Radboud training WSIs. Thus, we create one large multi-source dataset encompassing better sample

		GG wF1	ISUP κ^2
$\mathscr{G}$	NIC(Tellez et al., 2019a)	44.0±1.0	45.7±2.4
	CLAM(Lu et al., 2021b)	66.3±1.0	**78.1**±**1.5**
	WHOLESIGHT (Graph)	**67.1**±**0.9**	77.4±1.2

Table 7.3 – Classification results on Karolinska dataset. The best performances for using image-level supervision are highlighted in **bold**.

Annot.		Radboud			Karolinska		Sicap		
	Method	avg. Dice	GG wF1	ISUP κ^2	GG wF1	ISUP κ^2	avg. Dice	GG wF1	ISUP κ^2
$\mathscr{C}$	WHOLESIGHT (Multiplex, NC)	64.8±0.6	58.5±1.4	74.0±1.5	67.6±1.4	78.8±1.2	55.8±0.6	75.0±3.9	92.8±3.0
$\mathscr{G}$	NIC(Tellez et al., 2019a)	-	27.6±5.0	40.6±7.2	43.1±2.4	45.0±4.7	-	27.3±6.3	36.1±9.1
	CLAM(Lu et al., 2021b)	-	**57.6**±**2.3**	**73.8**±**2.3**	65.5±1.3	77.3±2.8	-	56.4±2.7	75.0±7.5
	WHOLESIGHT (Graph, GRAD-CAM)	29.0±1.2	56.5±0.5	72.0±1.5	**68.1**±**0.6**	**77.4**±**0.9**	24.2±2.1	**64.2**±**4.7**	**86.9**±**4.4**
	WHOLESIGHT (Graph + Pseudo, NC)	**46.0**±**0.4**	56.5±0.5	72.0±1.5	**68.1**±**0.6**	**77.4**±**0.9**	**41.6**±**0.5**	**64.2**±**4.7**	**86.9**±**4.4**
Bayes	WHOLESIGHT-MCD	43.9±1.8	58.2±0.8	73.7±3.1	67.9±1.1	77.7±1.0	44.5±3.0	61.4±3.6	75.2±6.7
	WHOLESIGHT-DE	46.3±0.2	60.6±0.6	76.5±0.7	68.6±0.4	78.1±0.6	46.6±1.7	66.0±1.5	84.5±1.2

Table 7.4 – Classification and segmentation results on Radboud, Karolinska, and Sicap datasets for models trained using both Radboud and Karolinska datasets.

variability and more diagnostically challenging cases than their standalone counterparts. The trained models on this curated dataset are tested individually on the Karolinska and Radboud test WSIs, herein studying the *in-domain* performance. Further, we test on the entire Sicap dataset, which constitutes of *out-of-domain* WSIs.

Performance analysis

Table 7.4 compares the classification performance of WHOLESIGHT, its Bayesian variants WHOLESIGHT-MCD and WHOLESIGHT-DE, CLAM, and NIC. For the Gleason grade weighted-F1 metric on the *in-domain* Karolinska and Radboud datasets, WHOLESIGHT reaches a comparable performance with respect to CLAM, and significantly outperforms NIC. Similar observations have prevailed for the ISUP κ^2 metric for both the *in-domain* datasets. However, the variances of Gleason grade weighted-F1 and ISUP κ^2 of the CLAM models are much higher than WHOLESIGHT. For testing on the *out-of-domain* Sicap dataset, WHOLESIGHT achieves significantly better Gleason grade weighted-F1 and ISUP κ^2 compared to competing CLAM and NIC methods. Even though the WHOLESIGHT variance on Sicap is larger compared to Karolinska and Radboud, it remains significantly lower than CLAM and NIC.

WHOLESIGHT-MCD performs comparable to WHOLESIGHT, without highlighting a clear performance gain for any of the datasets. Further, the variances of WHOLESIGHT-MCD meth-

ods are significantly higher than standalone WHOLESIGHT. However, WHOLESIGHT-DE shows a significant gain in classification and segmentation performances for all datasets. The deep ensemble-based methods result in clear advantages over MC-dropout-based methods, which are consistent with the observations by Thagaard et al. (2020). Noticeably, the gain in performances is higher on the *out-of-domain* dataset, compared to *in-domain* datasets. This finding corroborates the conclusion of Gustafsson et al. (2019) which showed that deep ensemble improves generalization to unseen cohorts. Overall, WHOLESIGHT-DE is the best performer across all datasets for all the evaluation metrics. Figure 7.3 presents the Gleason grading confusion matrices of WHOLESIGHT-DE on the three considered datasets. It can be observed that most misclassifications lie close to the diagonal. The majority of the confusion occurs between GG6 and GG7, *i.e.*, GG(3 + 3) versus GG(3 + 4) and GG(4 + 3). Such ambiguity is prevalent among pathologists, as presented in Ozkan et al. (2016); Salmo (2015). Further confusion matrices for Gleason grading, ISUP grading, primary classification, and secondary classification are presented in Figure D.3.

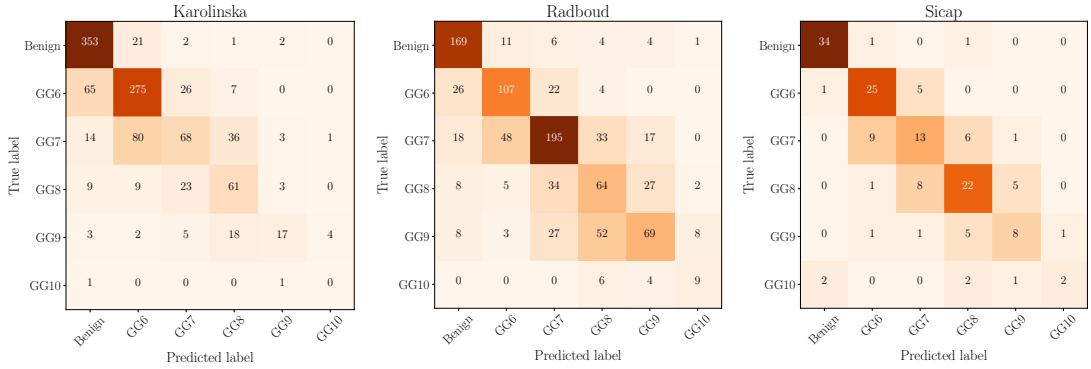


Figure 7.3 – Confusion matrix of Gleason grade classification for the WHOLESIGHT-DE method on the Karolinska, Radboud, and Sicap datasets.

Table 7.4 also presents the generalizability assessment of segmentation for WHOLESIGHT, and its variants WHOLESIGHT-MCD, WHOLESIGHT-DE on Radboud and Sicap datasets. Both WHOLESIGHT-MCD and WHOLESIGHT-DE significantly outperform the WHOLESIGHT method by improving the mean Dice score by +2.9% and +5.0%, respectively. Consistently with the observations for classification, WHOLESIGHT-DE is the best performer in terms of class-wise and aggregated Dice scores and systematically reduces the model performance variance. Benign regions, being the most common class in the dataset, reaches the highest Dice score. Whereas the less encountered Gleason patterns, *i.e.*, G3, G4, G5, have comparatively lower Dice scores. The drop in the Dice scores for these patterns primarily occurs due to the ambiguities among the cancerous patterns and false positive benign regions.

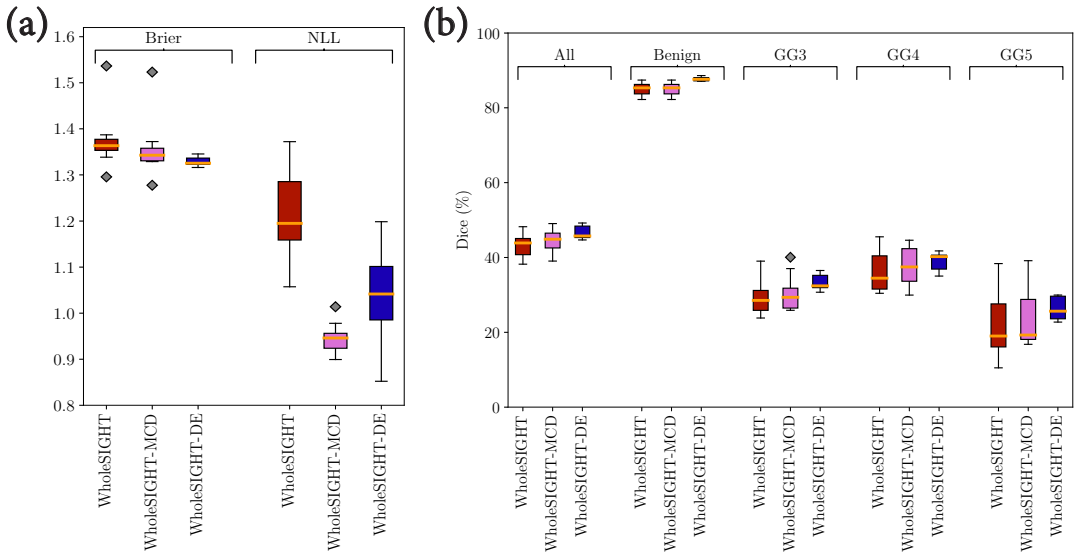


Figure 7.4 – (a) Uncertainty analysis of WHOLESIGHT, WHOLESIGHT-MCD and WHOLESIGHT-DE in terms of Brier and NLL metrics on the Sicap dataset. (b) Average and per-class Dice scores obtained on the Sicap dataset.

Uncertainty estimate analysis

Figure 7.5 presents the classification uncertainty analysis of the WHOLESIGHT, WHOLESIGHT-MCD, and WHOLESIGHT-DE methods, in terms of NLL (Figure 7.5(a)) and Brier score (Figure 7.5(b)), on Karolinska, Radboud and Sicap. The Bayesian methods, *i.e.*, WHOLESIGHT-MCD, and WHOLESIGHT-DE, render a significantly lower NLL than WHOLESIGHT across all datasets, for primary, secondary, and Gleason grade (P+S) classification. The relative gain of WHOLESIGHT-DE is +34.1% for P+S on Karolinska, +44.71% on Radboud, and +51.59% on Sicap. Interestingly, the gain is higher for the *out-of-domain* dataset, showing that Bayesian models, in particular deep ensembles, provide better uncertainty estimates. These observations are also consistent with the Brier score. WHOLESIGHT-DE consistently outperforms WHOLESIGHT, with a relative gain of +13.37% on Karolinska, +15.45% on Radboud, and +21.87% on Sicap. Noticeably, the NLL and Brier scores are consistently higher for predicting the secondary Gleason patterns compared to the primary patterns. This resonates with the fact that identifying secondary patterns is a more challenging task with higher ambiguity.

A similar analysis for quantifying the uncertainty in segmentation for WHOLESIGHT, WHOLESIGHT-MCD, and WHOLESIGHT-DE methods in terms of NLL and Brier scores on Sicap dataset, is presented in Figure 7.4(a). A relative gain of +21.49% and +1.44% in NLL and Brier score, respectively, is achieved by WHOLESIGHT-MCD on average Dice metric. Though WHOLESIGHT-DE outperforms WHOLESIGHT-MCD in terms of NLL, it performs inferior in terms of Brier score.

Model calibration analysis:

A model with a good uncertainty estimate should be well-calibrated, *i.e.*, the model confidence should be close to the underlying model performance. Figure 7.5(c) presents the reliability diagrams of the primary classification head on Karolinska and Radboud datasets. WHOLESIGHT-DE shows significantly better calibration than WHOLESIGHT-MCD and WHOLESIGHT-DE in accordance with the uncertainty estimate analysis. However, we observe that WHOLESIGHT-DE remains over-confident as the model accuracy (in orange) is lower than the expected optimal calibration (in blue). Figure 7.6 shows a detailed analysis of model calibration. We observe that even if not perfectly aligned, the gap between model accuracy and model confidence, denoted as dashed vertical lines in black, is reduced for the Bayesian methods. This gain is quantified by computing the ECE. For instance, the Radboud secondary classification head calibration is improved by +27.7% for WHOLESIGHT-MCD and +46.4% for WHOLESIGHT-DE.

7.4.6 Qualitative analysis

We qualitatively analyze the results of our proposed WHOLESIGHT method by (i) visualizing overlaid segmentation masks on WSIs, (ii) analysing the t-distributed stochastic neighbor (t-SNE) (Van der Maaten and Hinton, 2008) node embeddings, and (iii) correlating the segmentation outputs with pathological reasonings.

Visualizing WHOLESIGHT segmentation masks

Figure 7.7 demonstrates segmentation predictions obtained with WHOLESIGHT and its variant, WHOLESIGHT(Multiplex, NC), on Sicap dataset. We can observe that WHOLESIGHT correctly delineates the cancerous regions in the WSIs. Zooming into different regions conclude that the tissue regions of TG, *i.e.*, the nodes of TG, (outlined in black in Figure 7.7) encode meaningful units of *homogeneous* tissue. It substantiates the relevance of using TG representations for segmenting the tissue regions into Gleason patterns. We further notice that WHOLESIGHT, in a few cases, predicts benign regions adjacent to cancerous patterns as cancerous. For example, the benign region, primarily consisting of stroma, in Figure 7.7(c) is predicted as G5. We argue that these false positive detections do not inhibit the applicability of the method, as neighboring cancerous regions are correctly detected. In a few other cases, WHOLESIGHT correctly detects missed cancerous regions in the ground truth annotations. For instance, in Figure 7.7(b), the missing G4 region in the upper part of the WSI is correctly identified by WHOLESIGHT.

On comparing with WHOLESIGHT with WHOLESIGHT(Multiplex, NC), we observe that several false positives are removed, *e.g.*, in Figure 7.7(a), thereby offering more accurate segmentation outputs. However, the improvements by WHOLESIGHT(Multiplex, NC) are achieved at the cost of training with pixel-level annotations that are hardly available in real-world practice. Thus, WHOLESIGHT appears to be an appealing compromise between segmentation

performance and annotation requirement for Gleason pattern segmentation.

Visualizing tissue-level t-SNE feature space

A t-SNE visualization of the learned tissue-level embeddings is demonstrated in Figure 7.8 for Sicap dataset. The t-SNE algorithm projects the GNN node embeddings onto a two-dimensional feature space, allowing to analyse the connection between node embeddings and the Gleason pattern distribution.

Figure 7.8(a) displays the t-SNE feature space for the *correctly* classified nodes, which highlights demarcated clusters for each Gleason pattern. The large cluster of benign nodes indicates the diversity of the benign tissue regions. Several patches from each Gleason pattern cluster are presented in Figure 7.8(d). We can observe the reduced nuclei differentiation across the patches from benign to Gleason grade 5. Further, Figure 7.8(b) and (c) display the t-SNE feature space for the misclassified nodes. Specifically, Figure 7.8(b) represents the ground truth node labels, and Figure 7.8(c) the predicted node labels. Different embedding locations are further selected and highlighted by different colored rectangles and put in relation with corresponding patches to indicate the inter-class ambiguities, as demonstrated in Figure 7.8(e). For example, the first row in Figure 7.8(e) showcases patches that are benign but are predicted as Gleason pattern-3. We can visually compare these patches with the Gleason pattern-3 patches in the third row of Figure 7.8(d). Similar ambiguities between other pairs of Gleason patterns are also included in Figure 7.8(e).

Interpreting model outcomes via predicted segmentations

Predicted segmentations provide human-understandable *interpretability* maps. For researchers, the segmentations allow to, (i) identify morphological patterns responsible for the WSI classification, (ii) analyse failure cases by inspecting the pixel-level predictions, and ultimately (iii) better understand the model behavior towards biomarker discovery. For pathologists, they assist to, (i) put in relation the predicted WSI-level Gleason scores and the highlighted pixel-level Gleason patterns, (ii) confirm that the morphology of the identified cancerous regions align with the pre-established diagnosis criteria.

Additionally, in the perspective of developing AI-assisted human-in-the-loop tools, a Gleason grading system that can simultaneously *classify* and *segment* WSIs is closer to the latest pathological standards. Indeed, recent revisions of the Gleason grading system (Epstein et al., 2014) emphasized the importance of reporting the percentage of each grade for better patient stratification and treatment selection (Cheng et al., 2007; Huang et al., 2014; Choy et al., 2016; Sharma et al., 2020). These percentages can be trivially derived from the predicted segmentation maps by counting the number of pixels belonging to each pattern. Naturally, such information is not available in mere WSI classification systems. Reporting per-grade percentage is particularly important in ambiguous and borderline cases. For instance, consider

two patients with Gleason score 3+4. When a small percentage of pattern-4 is present, *e.g.*, 10%, the case can be considered as an intermediate risk cancer where active patient surveillance is enough (Amin et al., 2014). However, a larger secondary pattern may require specific treatments. Reporting percentages of each grade allows us to discriminate between these two scenarios easily.

Similarly, consider a Gleason score 4+3 with a small secondary Gleason pattern, *e.g.*, 90% and 10% area for primary and secondary patterns, respectively. This case will be scored as 4+3, even though it is close to a score of 4+4, which would lead to a different treatment protocol. By explicitly reporting the Gleason pattern percentages, such corner cases can be avoided.

7.5 Conclusion

Accurate delineation of patterns in a giga-pixel sized whole-slide histopathology image by using a deep learning method typically demands pixel-level annotations. However, such exhaustive annotations are often impossible to acquire in a real-world scenario due to time, effort, and expense bottlenecks. Nonetheless, the semantic segmentation of diagnostically relevant patterns is crucial for disease diagnosis and treatment selection. To this end, we have proposed a novel weakly-supervised semantic segmentation method, WHOLESIGHT, that can segment the relevant patterns of interest in histopathology images by leveraging only image-level supervision. To the best of our knowledge, WHOLESIGHT is the first weakly-supervised semantic segmentation method that can operate in an end-to-end manner on histopathology images of arbitrary shape and size. First, WHOLESIGHT transforms a histopathology image into a tissue-graph representation, where the nodes and edges of the graph denote tissue regions and tissue-to-tissue interactions. Second, the method employs a graph neural network to construct inter-tissue relationship-aware representations for the tissue regions. These contextualized representations are further used to classify the tissue-graph. Subsequently, pseudo-labels are generated for the tissue regions via a graph-feature-attribution technique, which enables the classification of the tissue regions and segments the input histopathology image. We evaluated our proposed method on three publicly available prostate needle biopsy datasets for Gleason grade classification and the delineation of different Gleason patterns in the biopsies. On comparing with state-of-the-art methods for histopathology applications, we demonstrated the classification and segmentation superiority of our proposed WHOLESIGHT method. Furthermore, we conducted extensive experimentation to assess the generalizability of WHOLESIGHT on *out-of-domain* histopathology datasets. In addition, we proposed a Bayesian extension of WHOLESIGHT, *i.e.*, WHOLESIGHT-DE, to enhance the generalizability of the method to images from different data sources. The generalizability is quantified in terms of classification and segmentation performance metrics, uncertainty estimation, and model calibration analysis. Notably, the proposed WHOLESIGHT method can utilize both image-level and pixel-level supervision to simultaneously perform image classification and segmentation tasks. Hence, WHOLESIGHT performance on both tasks can be enhanced in the presence of pixel-level partial annotations from pathologists. Though we have evaluated our

method for H&E stained prostate cancer needle biopsies, the technology is easily extendable to other tissue types, *e.g.*, breast, colon, lungs, etc., imaging techniques, *e.g.*, tissue microarrays, resection biopsies, etc., and image modalities, *e.g.*, other staining types in histopathology, multiplexed histopathology images, etc., and domains, *e.g.*, natural images, hyperspectral images, satellite images, other medical imaging domains, etc.

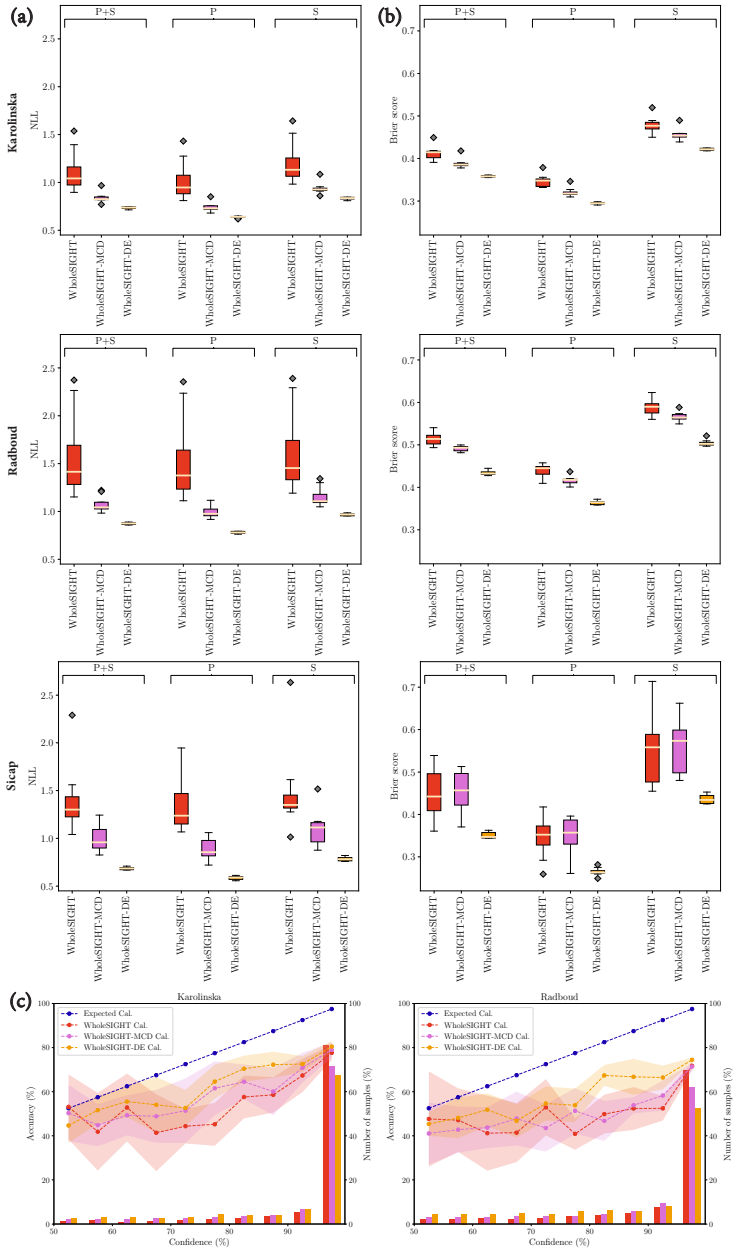


Figure 7.5 – Uncertainty analysis of the proposed WHOLESIGHT, WHOLESIGHT-MCD, and WHOLESIGHT-DE models. Each model was trained by combining Karolinska and Radboud train sets, and subsequently individually tested on Karolinska and Radboud test sets and the entire Sicap dataset. (a) Brier analysis (lower is better) on Karolinska, Radboud and Sicap. (b) NLL analysis (lower is better) on Karolinska, Radboud and Sicap. (c) Reliability diagrams on Karolinska and Radboud test sets for the primary Gleason classification head. The expected calibration (blue) highlights a perfectly calibrated model, where the performance in each bin matches the probability confidence. Calibrations of WHOLESIGHT, WHOLESIGHT-MCD, and WHOLESIGHT-DE are highlighted in red, purple, and orange, respectively. The number of samples (in %) in each bin is shown in red, purple and orange, respectively.

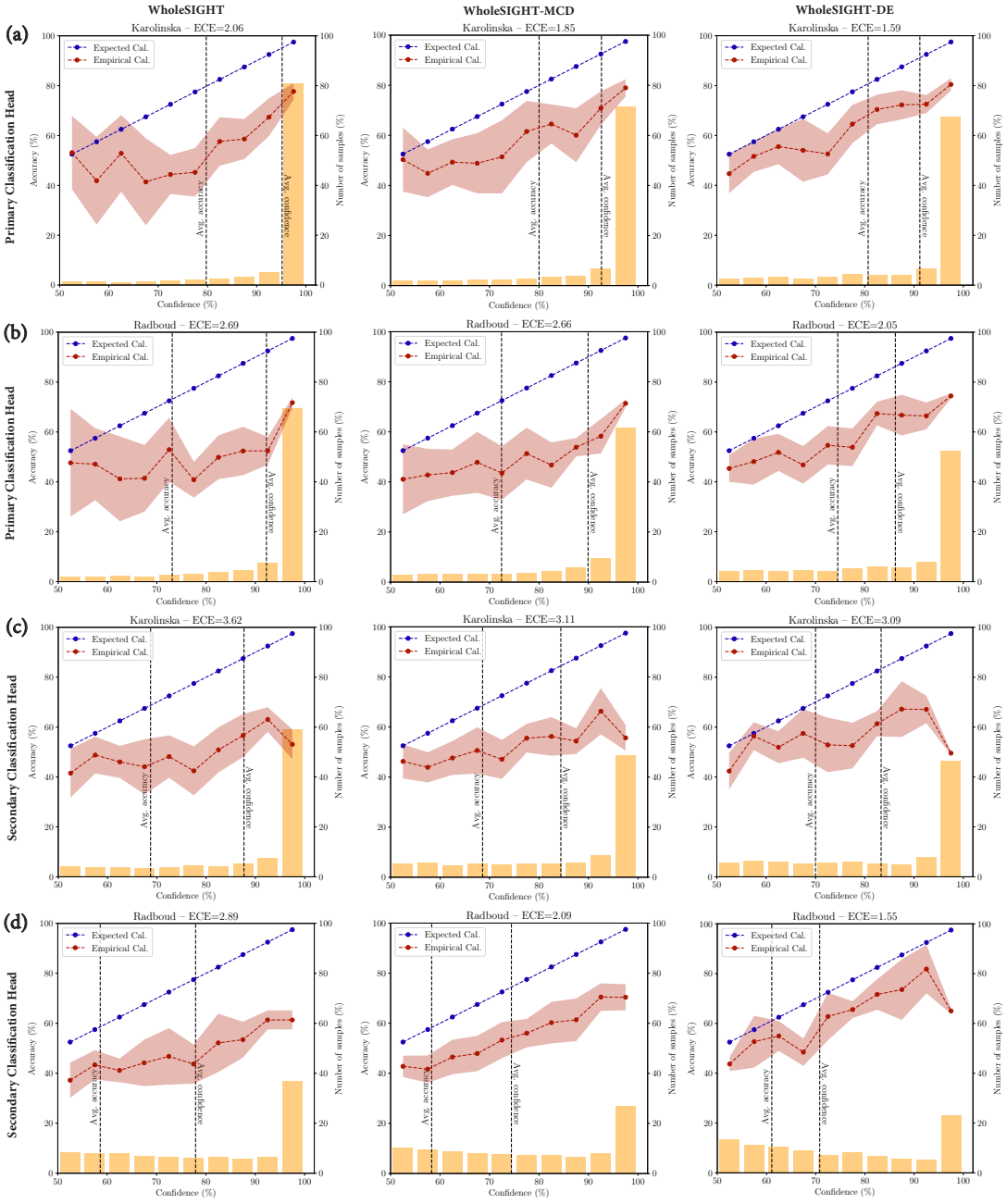


Figure 7.6 – Reliability diagrams of WHOLESIGHT, WHOLESIGHT-MCD, and WHOLESIGHT-DE tested on Karolinska and Radboud datasets for the primary and secondary Gleason classification heads. The expected calibration (blue) highlights a perfectly calibrated network, where the performance matches the probability confidence of the network. The observed network calibrations are highlighted in red. The number of samples (in %) in each classification bin is shown in orange. (a) Primary classification calibration on Karolinska test set. (b) Primary classification calibration on Radboud test set. (c) Secondary classification calibration on Karolinska test set. (d) Secondary classification calibration on Radboud test set.

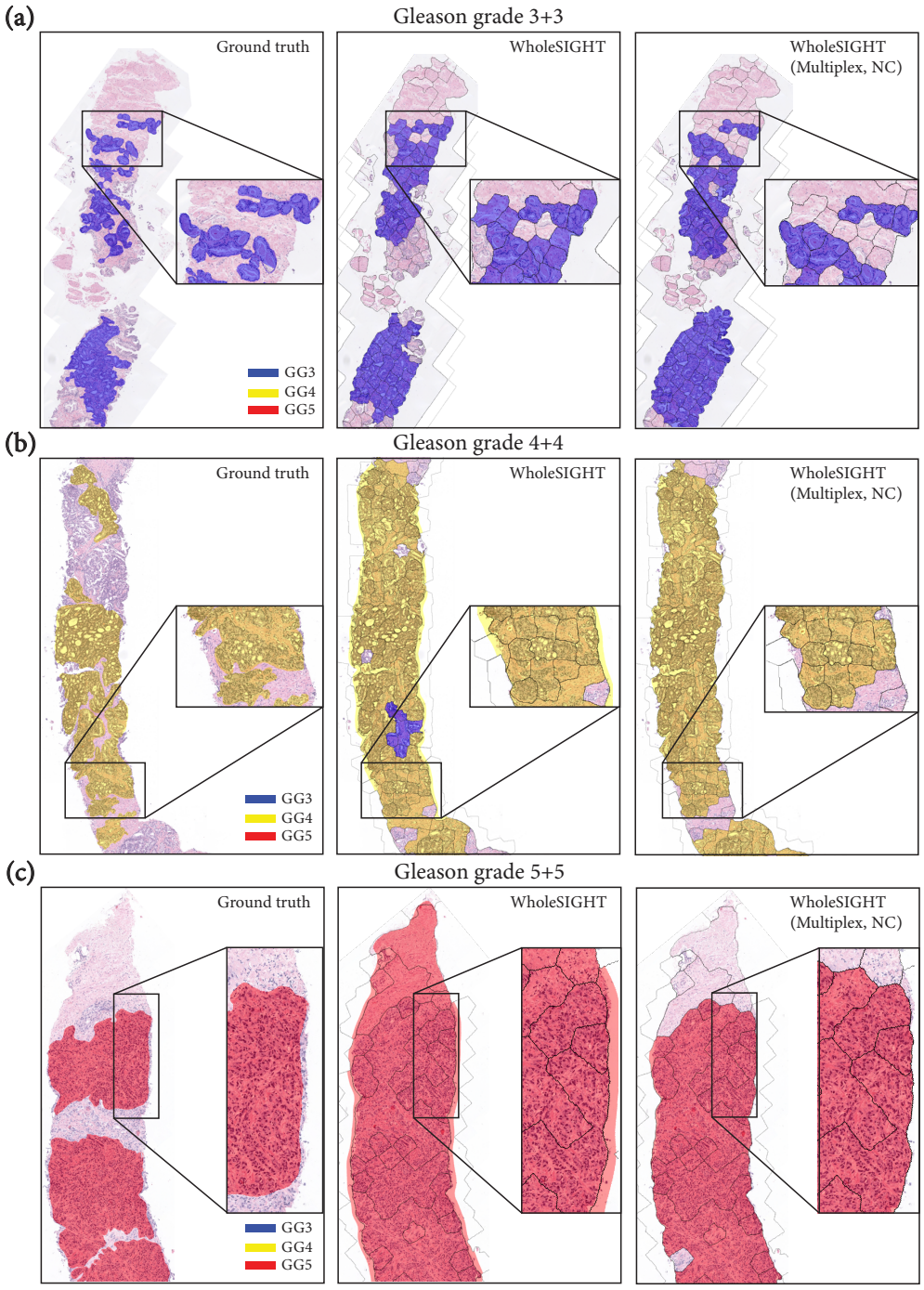


Figure 7.7 – Example of segmentation maps from the Sicap dataset. The Ground truth is shown in the left column, our proposed WHOLESIGHT in the middle column, and WHOLESIGHT(Multiplex, NC) in the right column. The tissue regions, *i.e.*, TG nodes, are represented by a black overlay. (a.) Example of a GG(3+3) sample. (b.) Example of a GG(4+4) sample. (c.) Example of a GG(5+5) sample. For better visualization, the benign areas are not represented in the segmentation maps.

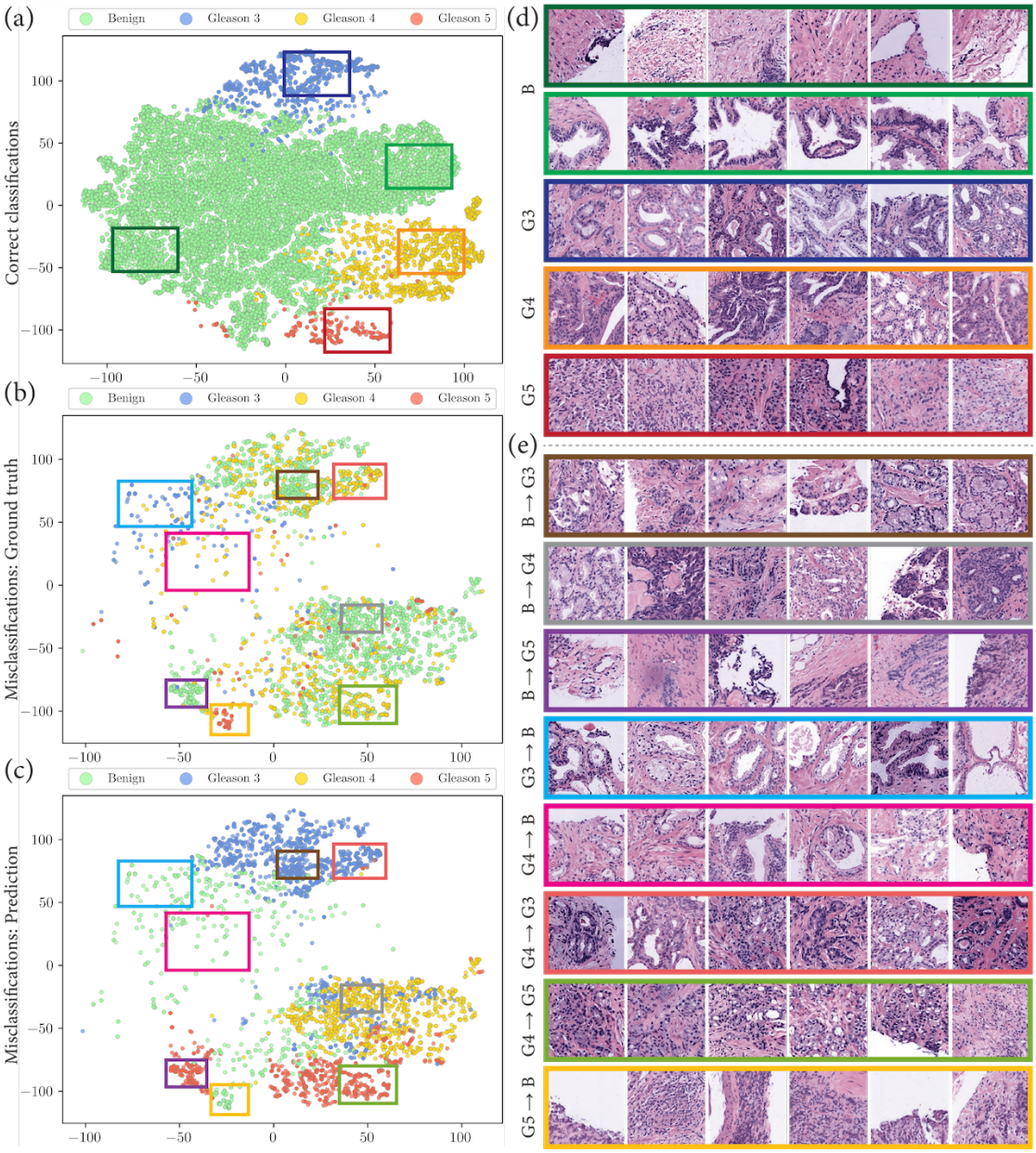


Figure 7.8 – t-SNE visualization of node-level feature representations and example patches corresponding to several regions on the two-dimensional t-SNE feature space for tissue-graphs in Sicap dataset. (a) t-SNE visualization of correctly classified nodes. (b) and (c) display the t-SNE visualization of misclassified nodes, where (b) and (c) highlight the ground truth and predicted class labels of the nodes, respectively. (d) and (e) demonstrate square patches of size 224 × 224 at 10× magnification cropped around the node centroids selected from different regions on the t-SNE embedding space. (d) and (e) highlight the correctly and incorrectly classified node patches, respectively. The labels of the patches in (e) are formatted as $Y \rightarrow \hat{Y}$, where Y and $\hat{Y}$ denote the ground truth and the predicted class labels. The colored rectangles around the patches in (d) and (e) correspond to respective colored rectangles in (a), (b), and (c).

Conclusion

In this chapter, we reformulate our contributions and main findings to highlight their strengths and limitations. Then, we present a set of future research directions, both on the methodological and clinical sides.

Discussion and limitations

The core idea of this thesis is to represent histology images as *entity-graphs* where nodes represent biological entities and edges interactions between these entities. This view is a complete paradigm shift from the traditional CNN-based processing of histology images, which are based on patch-level processing and aggregation. We showed that a number of challenges encountered in traditional approaches, *e.g.*, context–resolution trade-off, optimal patch prediction aggregation, and multi-scale information extraction fusion, can be addressed using our proposed *entity-graph* processing framework. *Entity-graphs* are built in three steps, namely (i) entity selection and detection to form the nodes of the graph, (ii) entity encoding to characterize the nodes, *e.g.*, using deep features extracted from a ResNet network or handcrafted morphological features, and (iii) a graph topology builder to define the edges, *e.g.*, k-NN graph. *Entity-graphs* are further processed by a GNN that can take different forms depending on the application at hand, *e.g.*, single-level PNA- or GIN-based GNNs, hierarchical GNN, etc.

Entity-graph processing brings scalability as GNNs can operate on arbitrary large graphs. To support this claim, we proposed the Hierarchical Cell-to-Tissue (HACT) representation that combines cell-level information with tissue-level information in a hierarchical fashion. Cell-level information is encoded as a cell-graph, where nodes are nuclei and edges nuclei-nuclei interactions. Tissue-level information is represented as a tissue-graph, where nodes are tissue regions and edges connect adjacent regions. To evaluate HACT, we created the BRACS dataset, the largest cohort to date of H&E histopathologic images for breast tumor classification. HACT combined with HACT-Net, a novel GNN designed for pathology, led to state-of-the-art results for breast tumor RoI classification showing performance superior to CNN methods and pathologists. While promising, this approach still suffers from several limitations. First, on the modeling side, HACT-Net is not trained end-to-end. It relies on pre-trained CNN models to encode the entity, *e.g.*, ImageNet features with a ResNet backbone

network. In its current shape, HACT-Net *could* in theory be trained end-to-end but at a really high computational cost, as the graphs would need to be built on-the-fly in the batch constructor. This limitation is common to most of the WSI-level approaches, *e.g.*, (Tellez et al., 2019a; Shaban et al., 2020; Lu et al., 2021b; Campanella et al., 2019). Some methods are using pathology-related auxiliary tasks to pre-train the feature extractor, thereby reducing the domain gap between pre-trained and downstream data (Tellez et al., 2019a). Other methods proposed unsupervised pre-training based on GANs and VAEs (Tellez et al., 2019a; Shaban et al., 2020). However, the gain compared to ImageNet pre-trained features remains marginal, if not nonexistent. Another important aspect is to understand *what* is making the entity-graph approach working, is it the structural entity-centric tissue decomposition, the entity selection, or the graph connectivity patterns? We hypothesize that any topology enforcing spatial connectivity would lead to similar performance, *e.g.*, k-NN, radius graph, etc. This is a reasonable assumption when the graphs are *homophilous*, *i.e.*, when your neighbors are likely to have the same functional property as you. This property is found in cell-graphs, where all the nuclei located in the gland (epithelial nuclei) will share morphological features, and nuclei outside the gland (*e.g.*, stromal nuclei or lymphocytes) will have different ones. In this sense, cell-graph GNNs can be seen as inducing a low-pass filtering over the nodes features, that learn to represent discriminative nuclei phenotypes. A similar reasoning can hold for tissue-graphs. When scaling to WSI-level, an unexplored tissue representation would be to use glands as entity. A WSI can include hundreds of glands that convey different information about the tissue. A major drawback of this representation is to develop a gland detector algorithm, an unexplored and non-trivial task.

As studied in Chapter 6, entity-based analysis preserves the notion of histopathological entity, which the pathologists can relate to and reason with. A main contribution of this thesis are post-hoc explainability tools to provide pathologist-friendly explanations. Our approach is using graph explainers to compare the focus of the model with prior pathological knowledge. While our study concluded that the salient regions highlighted by the model were relevant, there exist a number of limitations. First, what is important for a pathologist is not obviously important for another one. Therefore, gathering a universal prior is not straightforward. Our intuition is that attribute-wise histograms are the most reliable evidences of the relevance of a given explainer to quantify the importance of a concept. An alternative when explaining a prediction to a pathologist would be to provide per-attribute cursors, showing the average value of the most important nuclei, normalized by some representative set. Another limitation is that this analysis relies on domain-specific knowledge, that requires a deep understanding of the task that only pathologists can provide. The main benefit of this method is also its main limitation, it remains task-specific and require work to be adapted to other tasks, even in related domains.

The development of scalable and explainable approaches is pivotal. However, they require annotations that are scarce and expensive to acquire. Another line of research developed in this thesis is concerned with weakly-supervised learning. We proposed WHOLESIGHT, an algorithm that can perform *segmentation* and *classification* with training only from WSI-level

labels. The core principle is to encode a WSI as a tissue-graph, to which we associate a label. First, a GNN is trained for graph classification, followed by a post-hoc node-level feature attribution to derive pseudo-labels, that are further used for node classification. The main drawback of this approach is that tissue component detection is both a time-consuming and sub-optimized process, that relies on basic image processing (SLIC superpixel). Ideally, a dedicated tissue component detector would be employed but it would require expensive annotations. Despite this limitation, WHOLESIGHT brings valuable complementary information with the addition of a segmentation head.

Future work

A thesis is a never-ending job, as new projects are completed, a plethora of new ideas emerge, some of which prove to be promising. Below we list some research directions and project ideas that could be explored, if time, pathological expertise, and data availability were not limitations.

Leverage pathology reports: First, weakly supervised algorithms have not yet expressed their full potential in computational pathology. Zhang et al. (2017) showed that pathology reports contain rich and complex information that is ready to be exploited. A non-exhaustive list of extractable information comprises cancer staging information, *e.g.*, tumor size, cancer grading information, *e.g.*, microscopic tumor description, tumor grade, patient identification information, *e.g.*, age, sex, ethnicity, and treatment response. This information can serve as training targets for multi-task classification, regression or clustering of WSIs. Alternative training signals can be extracted in other modalities *e.g.*, radiology, genomics as proposed in Chen et al. (2020). Following this approach, we reduce the annotation effort required by pathologists, allowing more time to be spent on testing and analysis of AI behaviour. *Entity-graph* processing can be leveraged as a backbone method to build WSI embeddings, *e.g.*, with WHOLESIGHT. The benefits of such training setting are three-fold. First, as some targets are *correlated*, the overall training signal is stronger (keeping in mind the large input size). Second, we can expect tasks to regularize each other. Indeed, *correlated* and *complementary* targets can bring valuable information to each other, hence reducing inter- and intra-observer variability, a notable challenge in pathology. Third, by training AI systems with multiple *interpretable* and *correlated* targets, *e.g.*, tumor grade and microscopy description, we can explore the *consistency* of trained algorithms. For instance, an indicator of trustworthiness would be a sample correctly classified as cancerous along with a predicted tumor description matching the expected appearance of cancerous tumor regions. This idea could be further explored by designing a consistency score based on prior pathological knowledge that would evaluate the coherence of a multi-task prediction. This score can be used to detect out-of-distribution samples in case of distribution shift, and therefore, improve model uncertainty prediction. The main constraint of implementing such project relies in data availability: gathering a large-scale WSIs dataset with associated *digitized* pathology report is hard. Even if public dataset size is increasing (from 400 samples in BACH (Aresta et al., 2019), to 4,000 BRACS (Pati

et al., 2021b), and 10,000 in PANDA (Bulten et al., 2020)), we are still far from having access to large-scale, multi-organ, annotated datasets. We can mention ongoing efforts in this direction like the IMI BigPicture project Moulin et al. (2021), a European initiative to build the largest public repository of WSIs.

Breast cancer biomarker discovery: In this thesis, we focused on building computer-aided diagnosis tools to develop both scalable and explainable methods. The field is reaching a mature enough state where several reliable DL-based algorithms exist to build WSI embeddings, *e.g.*, WHOLESIGHT. This technology can be leveraged towards the discovery of new cancer biomarkers. The implementation of such a project requires close collaboration with pathologists to identify a promising problem statement. For instance, in breast cancer, a cancer type denoted as HER2 positive, *i.e.*, if the patient tests positive to the human epidermal growth factor receptor 2 protein, is still not fully understood, and some phenotypical patterns remain unexplored. We know that the H&E modality encode information about HER2 status. A natural first step towards better HER2 positive understanding would be to predict the HER2 status from the H&E image directly. Then, by putting in relation H&E and HER2 modalities, we can study their inter-dependence. This can be achieved, for instance, by studying the important regions for predicting HER2 status given the HER2 and theH&E image, *e.g.*, with a post-hoc explainer. We can expect that some regions will be shared, while other ones, unexplored might bring novel insights.

A beginner's manual to pathology for computer scientists: This idea is not a research project per se, but rather a community service. As the name suggests, CompPath aims to apply computational methods to pathological data. The vast majority of the community is composed of computational experts, without dedicated pathological training. As a consequence, the required pathological knowledge is learned on-the-fly, with interactions with pathologists. This is leading to incomplete understanding of the pathological workflow and basic diagnosis principles. We argue that this situation can hinder the development of clinically relevant methods, and even push the community to focus on inapplicable and unrealistic problems. To address these concerns, we propose to write a beginner's manual to pathology for computer scientists that would include the minimum pathological knowledge to confidently start working in computational pathology. Such manuscript would cover (i) tissue acquisition and preparation, (ii) the different staining techniques and their usability, (iii) the main diagnosis, prognosis and treatment response biomarkers for certain cancers, (iv) some of the main cancer staging and grading systems, (v) a contextualized perspective of pathology in patient care and its integration in the medical workflow, *e.g.*, interactions with oncologists, radiologists, etc. (vi) an overview of the WSI-scanning devices. This work needs to be conducted in collaboration with pathologists and computer scientists, to connect the two fields.

A Class Activation Maps: Intuition and Justifications

A.1 Connection between CAM and GRAD-CAM

CAM, initially proposed by Zhou et al. (2016) aims to highlight important regions in images, when a CNN model with Global Average Pooling (GAP) followed by a softmax classifier is employed. GAP in CNN terminology is equivalent to a *mean* readout aggregation followed by a softmax classifier in a GNN.

Formally, a *mean* readout with a softmax layer can be written as:

$$y(c) = \sum_{k=1}^{d^{(T)}} w_{k,c}^{(T)} \sum_{n=1}^{|V|} H_{n,k}^{(T)} \tag{A.1}$$

where $y(c)$ denotes the logit value of the c^{th} class, $d^{(T)}$ is the node feature dimension at layer T, *i.e.*, equivalent to the number of channels in a CNN, $w_{k,c}^{(T)}$ is the k^{th} channel importance score of class c, and $H_{n,k}^{(T)}$ represents the k^{th} node feature of node n at layer T, *i.e.*, at the last layer. We can further decompose Equation A.1 as:

$$F_k = \sum_{n=1}^{|V|} H_{n,k}^{(T)} \tag{A.2}$$

$$y(c) = \sum_{k=1}^{d^{(T)}} w_{k,c}^{(T)} F_k \tag{A.3}$$

where we introduce the notation for the aggregated node features as F_k.

Computing the gradient of $y(c)$ w.r.to F_k, and applying the chain-rule, we get:

$$\frac{\partial y(c)}{\partial F_k} = \frac{\frac{\partial y(c)}{\partial H_{n,k}^{(T)}}}{\frac{\partial F_k}{\partial H_{n,k}^{(T)}}} \tag{A.4}$$

Now computing the partial derivative of Equation (A.2) w.r.to $H_{n,k}^{(T)}$, as:

$$\frac{\partial F_k}{\partial H_{n,k}^{(T)}} = 1 \tag{A.5}$$

which we can combine with Equation (A.4) as:

$$\frac{\partial y(c)}{\partial F_k} = \frac{\partial y(c)}{\partial H_{n,k}^{(T)}} \tag{A.6}$$

By taking the partial derivatives of $y(c)$ w.r.to F_k (see Equation A.3), we obtain:

$$\frac{\partial y(c)}{\partial F_k} = w_{k,c}^{(T)} \tag{A.7}$$

By combining Equation (A.6) and Equation (A.7), we obtain:

$$w_{k,c}^{(T)} = \frac{\partial y(c)}{\partial H_{n,k}^{(T)}} \tag{A.8}$$

Finally, we can sum on both sides over the graph nodes to derive that:

$$\sum_{n=1}^{|V|} w_{k,c}^{(T)} = \sum_{n=1}^{|V|} \frac{\partial y(c)}{\partial H_{n,k}^{(T)}} \tag{A.9}$$

which leads to:

$$w_{k,c}^{(T)} = \frac{1}{|V|} \sum_{n=1}^{|V|} \frac{\partial y(c)}{\partial H_{n,k}^{(T)}} \tag{A.10}$$

We obtain the GRAPHGRAD-CAM formulation where we explain class c w.r.to the last (graph-)convolutional layer, as introduced in Chapter 3.

A.2 Derivation of channel-wise weights in GRAPHGRAD-CAM++

We modify GRAPHGRAD-CAM formulation as:

$$w_k^{(t)} = \sum_{n=1}^{|V|} \alpha_{n,k}^{(t)} \text{ReLU}\left(\frac{\partial y_{max}}{\partial H_{n,k}^{(t)}}\right) \tag{A.11}$$

where the weights $\alpha_{n,k}^{(t)}$ aim to better localize node contributions. It was shown that CAM (Zhou et al., 2016) have better localization properties than GRAD-CAM. Therefore, we would like to derive a closed-form solution for $\alpha_{n,k}^{(t)}$ based on the CAM formulation (see Equation (A.1)). By

combining Equation (A.1) and Equation (A.11), we obtain:

$$y(c) = \sum_{k=1}^{d^{(T)}} \Big(\sum_{n=1}^{|V|} \alpha_{n,k}^{(t)} \frac{\partial y(c)}{\partial H_{n,k}^{(t)}} \Big) \sum_{n=1}^{|V|} H_{n,k}^{(t)} \tag{A.12}$$

where we have dropped the ReLU activation for simplicity. By taking the partial derivative of $y(c)$ w.r.to $H_{n,l}^{(t)}$, we derive:

$$\frac{\partial y(t)}{\partial H_{n,k}^{(t)}} = \sum_{n=1}^{|V|} \alpha_{n,k}^{(t)} \frac{\partial y(c)}{\partial H_{n,k}^{(t)}} + \sum_{n=1}^{|V|} H_{n,k}^{(t)} \Big(\alpha_{n,k}^{(t)} \frac{\partial^2 y(c)}{(\partial H_{n,k}^{(t)})^2}) \Big) \tag{A.13}$$

That can further be partially derived w.r.to $H_{n,k}^{(t)}$, as:

$$\frac{\partial^2 y(c)}{(\partial H_{n,k}^{(t)})^2} = 2\alpha_{n,k}^{(t)} \frac{\partial^2 y(c)}{(\partial H_{n,k}^{(t)})^2} + \sum_{n=1}^{|V|} H_{n,k}^{(t)} \Big(\alpha_{n,k}^{(t)} \frac{\partial^3 y(c)}{(\partial H_{n,k}^{(t)})^3}) \Big) \tag{A.14}$$

Finally leading to:

$$\alpha_{n,k}^{(l)} = \frac{\frac{\partial^2 y(c)}{(\partial H_{n,k}^{(l)})^2}}{2\frac{\partial^2 y(c)}{(\partial H_{n,k}^{(l)})^2} + \sum_{n=1}^{|V|} H_{n,k}^{(l)} \Big(\frac{\partial^3 y(c)}{(\partial H_{n,k}^{(l)})^3} \Big)} \tag{A.15}$$

We obtain that GRAPHGRAD-CAM++ channel-wise weights are computed as:

$$w_k^{(t)} = \sum_{n=1}^{|V|} \frac{\frac{\partial^2 y(c)}{(\partial H_{n,k}^{(t)})^2}}{2\frac{\partial^2 y(c)}{(\partial H_{n,k}^{(t)})^2} + \sum_{n=1}^{|V|} H_{n,k}^{(t)} \Big(\frac{\partial^3 y(c)}{(\partial H_{n,k}^{(t)})^3} \Big)} \mathrm{ReLU}\Big(\frac{\partial y(c)}{\partial H_{n,k}^{(t)}} \Big) \tag{A.16}$$

where $y(c)$ is typically set to $y_{\max}$.

B Open-source Implementations, Libraries and Reproducibility

The ideas, results, and implementations presented in this section are published in:

- "HistoCartography: A Toolkit for Graph Analytics in Digital Pathology", **Guillaume Jaume***, Pushpak Pati*, Valentin Anklin, Antonio Foncubierta, Maria Gabrani. In *International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI), Third MICCAI workshop on Computational Pathology*, 2021, (Jaume et al., 2021a).

GJ (the author of this thesis) is sharing first co-authorship with PP. The ideas and experiments were designed by GJ and PP. The development of the library was led by GJ and implemented by GJ, PP and VA. The experiments were conducted by GJ and PP. The manuscript was written by GJ and PP and subsequently revised by MG and AF.

B.1 Introduction

Publishing code and providing enough information so that experiments and results can be easily *replicated* are critical to advancing CompPath research. Notably, open-sourcing libraries with production-level code facilitates the comprehension of publications and accelerate the development of new methods. This is particularly true for the proposed entity-graph processing presented in Chapter 5,6,7 that demands several prerequisites, such as entity detection, entity encoding, constructing the graph topology etc., alongside standard preprocessing, such as stain normalization, tissue detection etc. Additionally, our proposed workflow requires to utilize recent advancements in DL for processing graph-structured data. All these obstacles may prevent the adoption of entity-graphs in CompPath. In addition, the lack of a standardized framework with the aforementioned functionalities urge the researchers to reinvent the wheel, which is cumbersome, time-consuming, hampers reproducibility, and requires a wide range of technical acumen.

To overcome these limitations, we introduce HISTOCARTOGRAPHY, an open-source python

library that facilitates graph-analytics in CompPath. Specifically, the contributions presented in this chapter are:

- A standardized python library that unifies a set of histology image manipulation tools, entity-graph builders, GNN models, and model explainability tools;
- A benchmark assessment of performance and scalability on classification and segmentation tasks in pathology;
- A review of extant libraries for histological image analysis.

B.2 Extant libraries in CompPath

Several open-source libraries have been proposed to develop CompPath pipelines. Most of them include helper functions to perform standard preprocessing and visualization. HISTOLAB (Arbitrio et al., 2020) includes WSI-level tissue detection and tile extraction modules. SYNTAX (Byfield et al., 2020) provides the same features with abstraction where modules can be stacked and run in a pre-defined pipeline. STAINTOOLS (Byfield et al., 2019) provides tools for stain normalization and augmentation. HISTOMICSTK (Beezley et al., 2021) enables to perform tissue detection, object detection and segmentation, image filtering, stain normalization and deconvolution, and handcrafted feature extraction. Further, HISTOMICSTK allows nuclei segmentation and classification using classical ML approaches. It also provides a UI to run containerized modules and pipelines. Though HISTOMICSTK includes valuable functionalities, it caters limited DL tools. Similarly, OPENSLIDE (Gilbert et al., 2020) provides a UI to read and visualize histology images that supports most of the WSI formats. Finally, QUPATH (Bankhead et al., 2021) offers a UI that allows to read, visualize and annotate WSIs. It also includes tools to perform stain normalization, nuclei and tissue detection, and implement basic ML models. However, QUPATH does not provide a python Application Programming Interface (API), which makes it difficult to integrate into existing workflow and DL frameworks, *e.g.*, PyTorch, Tensorflow. Most importantly, none of the frameworks provide graph-related helpers. With the advent of graph-techniques as a new paradigm for analyzing histology images, a standardized library is desired for reinforcing the development.

B.3 Histocartography: graph analytics tools for CompPath

In this section, we highlight the core functionalities of HISTOCARTOGRAPHY, namely (i) a *preprocessing* module that includes a set of histology image processing tools and entity-graph builders, (ii) an *ML* module with helpers to learn from entity-graphs, (iii) an *explainability* module, that includes a set of graph interpretability tools. The specific functionalities in each module are summarized in Table B.1.

Function	Module	Input	Output	Existing	CPU	GPU
Preprocessing	Vahadane Stain Norm	`I`	`I`	✓	✓	✗
	Macenko Stain Norm	`I`	`I`	✓	✓	✗
	Tissue Mask Detection	`I`	`M`	✓	✓	✗
	Nuclei Detection	`I`	`M`	✓	✓	✓
	Nuclei Concepts	`I`, `M`	`M`	✓	✓	✗
	Tissue Component Detection	`I`	`M`	✗	✓	✗
	Deep Feature Extraction	`I`, `M`	`X`	✗	✓	✓
	Feature Cube Extraction	`I`	`X`	✗	✓	✓
	k-NN Graph Building	`X`, `M`	`G`	✗	✓	✗
	RAG Graph Building	`X`, `M`	`G`	✗	✓	✗
ML	Cell-Graph Model	`G`	`P`	✗	✓	✓
	Tissue-Graph Model	`G`	`P`	✗	✓	✓
	HACT Model	`G`, `G`, `X`	`P`	✗	✓	✓
Explainers	GNNEXPLAINER	`G`	`S`	✗	✓	✓
	GRAPHGRAD-CAM	`G`	`S`	✗	✓	✓
	GRAPHGRAD-CAM++	`G`	`S`	✗	✓	✓
	GRAPHLRP	`G`	`S`	✗	✓	✓

Table B.1 – Overview of HISTOCARTOGRAPHY functionalities, with the i/o, CPU and GPU compatibility, and availability in extant libraries for individual module. `I`, `M`, `X`, `G`, `P` and `S` denote an image (np.array (Harris et al., 2020)), a mask (np.array), features (torch.Tensor (Paszke et al., 2019)), a graph (DGLGraph (Wang et al., 2019a)), predictions (torch.Tensor) and importance scores (torch.Tensor), respectively.

B.3.1 Preprocessing module

Stain normalization: Variation in H&E staining protocols for tissue specimens induces appearance variability that adversely impacts computational methods (Tellez et al., 2019b). To alleviate these variations, HISTOCARTOGRAPHY implements two popular normalization algorithms proposed by Macenko et al. (2009) and Vahadane et al. (2016), similar to STAINTOOLS and HISTOMICSTK, which supports both reference-based and reference-free normalization, *i.e.*, with manual stain vectors. Figure B.2 highlights a sample normalization output using our API and Figure B.1 presents a code snippet to implement Vahadane stain normalization following the HISTOCARTOGRAPHY syntax.

Tissue Detection: A WSI usually includes significant regions without tissue. Identifying the tissue regions can confine the analysis and reduce computational effort. The tissue detector in HISTOCARTOGRAPHY iteratively applies Gaussian smoothing and Otsu thresholding until the mean of non-tissue pixels is below a threshold. The simple, yet effective tool ensures speed and scalability. Figure B.1 presents the syntax for tissue detection in HISTOCARTOGRAPHY. This module is common across HISTOLAB, SYNTAX, HISTOMICSTK and QUPATH.

Nuclei detection: This module enables to segment and locate nuclei in H&E images. Though it is well-studied in CompPath, only a few public implementations are available. For instance,

Stain Normalization Code

```
from PIL import Image
import numpy as np
from histocartography.preprocessing import (
  VahadaneStainNormalizer
)

# define Vahadane stain normalizer
normalizer = VahadaneStainNormalizer(
  target_path='target.png'
)

# process image
img = np.array(Image.open('17B0031061.png'))
norm_image = normalizer.process(img)
```

Tissue Mask Detection Code

```
from PIL import Image
from histocartography.preprocessing import (
  GaussianTissueMask
)

# define tissue detector
tissue_detector = GaussianTissueMask(
  kernel_size=20,
  sigma=10
)

# process image
img = np.array(Image.open('17B0031061.png'))
mask = tissue_detector.process(img)
```

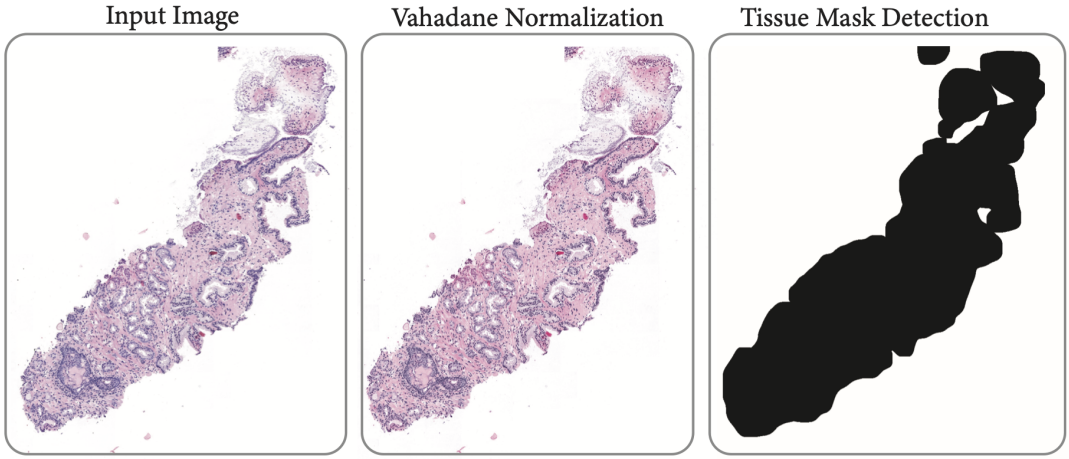


Figure B.1 – Implementation of Vahadane stain normalization (left) and tissue mask detection (right) with the *Preprocessing* functionalities in the HistoCartography API.

QuPath allows to detect nuclei but requires model training and fine-tuning. While providing flexibility, the module includes only elementary ML methods. HistoCartography integrates two checkpoints from the state-of-the-art HoVerNet model (Graham et al., 2019a) trained on PanNuke (Gamper et al., 2019) and MoNuSac (Ruchika et al., 2020) datasets for nuclei segmentation and classification. This module is used to build cell-graphs as presented in Chapter 5.

Tissue Component Detection: HistoCartography includes an unsupervised superpixel-based approach to segment tissue regions. First, the tissue is oversegmented into homogeneous superpixels using SLIC (Achanta et al., 2012) algorithm. Then, neighboring superpixels are hierarchically merged using color similarity to denote meaningful tissue regions, *e.g.*, epithelium and stroma regions. Superpixels depicting tissue regions are used by Bejnordi et al. (2015); Pati et al. (2020, 2021a) and thoroughly introduced in Chapter 5.

Feature Extraction: HistoCartography includes two types of feature extractors, *i.e.*, handcrafted- and CNN-based, to encode the entity characteristics.

The handcrafted feature extractor computes entity-level morphological and topological prop-

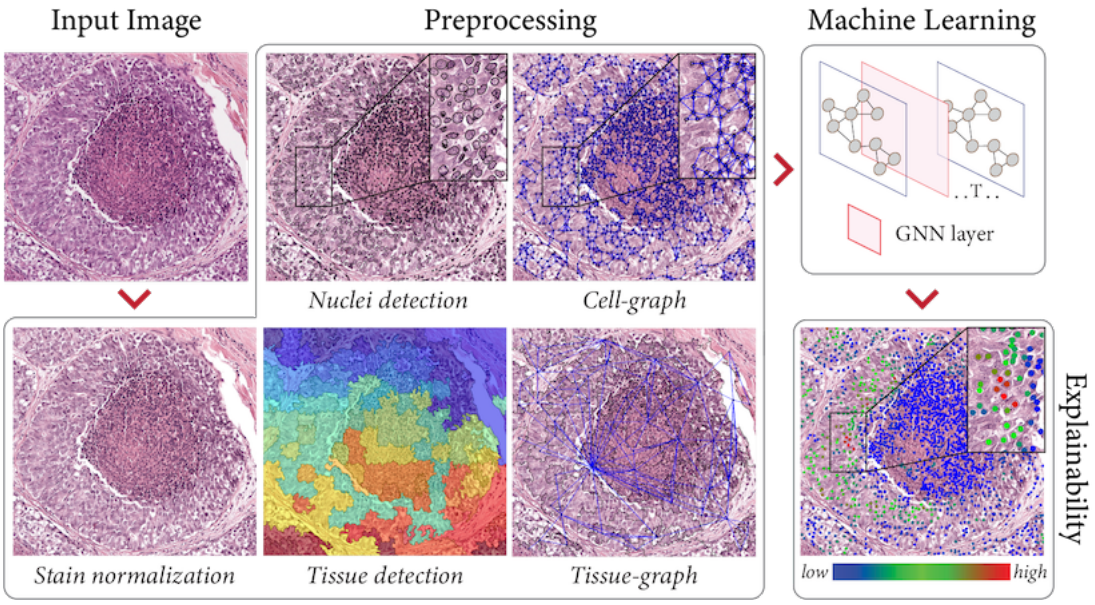


Figure B.2 – Overview of HISTOCARTOGRAPHY functionalities and modules.

erties. Morphological features include shape, size and texture properties, namely, entity area, convex area, eccentricity, equivalent diameter, euler number, length of the major and minor axis, orientation, perimeter, solidity, convex hull perimeter, roughness, shape factor, ellipticity, roudness. Texture properties are based on gray-level co-occurrence matrices (GLCM). Specifically, we extract the GLCM contrast, dissimilarity, homogeneity, energy, angular speed moment and dispersion. The topological features are based on the entity density computed as the mean and variance of entity crowdedness. Handcrafted features can be used for training DL algorithms (Demir et al., 2004; Zhou et al., 2019a; Pati et al., 2020; Studer et al., 2021), or concept-based post-hoc explainability as presented in Chapter 6.

The deep feature extractor allows to extract CNN features by using any pre-trained deep architecture, *e.g.*, ResNet, MobileNet, embedded in torchvision (Marcel et al., 2010). The module intakes patches centered around the entity and extracts features from the penultimate layer of the architectures. If the entity is larger than the specified patch size, then multiple patches within the entity, w/ or w/o overlapping, are processed, and the final feature is computed as the mean of the patch-level deep features, as used in Chen et al. (2020); Pati et al. (2020, 2021a) and Chapter 5,6,7. Deep features can alternatively be extracted from the WSI to build a feature-cube as suggested by Shaban et al. (2020); Tellez et al. (2019a).

Graph builders: HISTOCARTOGRAPHY presents two graph builders, *i.e.*, the thresholded k-NN and the RAG. The k-NN graph builder defines the graph topology by connecting each entity to its k-closest neighbors. Connections between distant entities beyond a threshold can be pruned to have spatial sparsity in the graph. We recommend this builder to connect single entities, *e.g.*, nuclei, glands. The RAG builder connects entities using spatial adjacency, *i.e.*, entities sharing a common boundary. It builds a sound topology when dealing with dense

segmentation maps, *e.g.*, tissue regions. Figure B.2 presents samples of cell- and tissue-graphs. Further, the module fuses the node features and the topological distribution to render a DGL graph for an image. Figure B.3 presents the code to implement a CG (left) and a TG with the HISTOCARTOGRAPHY API. Noticeably, these functionalities require only ten lines of code by using HISTOCARTOGRAPHY, which could have otherwise required a few hundred lines.

Cell Graph Generation Code

```python
from PIL import Image
from histocartography.preprocessing import (
  NucleiExtractor,
  DeepFeatureExtractor,
  KNNGraphBuilder
)

# define nuclei
nuclei_detector = NucleiExtractor()

# define feature extractor
feats_extractor = DeepFeatureExtractor(
    architecture='resnet34',
    patch_size=72,
    resize_size=224
)

# define graph builder
knn_graph_builder = KNNGraphBuilder(
  k=5,
  thresh=50
)

# process image
img = np.array(Image.open('image.png'))
nuclei, _ = nuclei_detector.process(img)
feats = feats_extractor.process(img, nuclei)
graph = knn_graph_builder.process(nuclei, feats)
```

Tissue Graph Generation Code

```python
from PIL import Image
from histocartography.preprocessing import (
    ColorMergedSuperpixelExtractor,
    DeepFeatureExtractor,
    RAGGraphBuilder
)

# define super-pixels
superpx_detector = ColorMergedSuperpixelExtractor(
    nr_superpixels=400,
    downsampling_factor=4
)

# define feature extractor
feats_extractor = DeepFeatureExtractor(
    patch_size=144,
    resize_size=224
)

# define graph builder
rag_graph_builder = RAGGraphBuilder()

# process image
img = np.array(Image.open('image.png'))
superpxs, _ = superpx_detector.process(img)
feats = feats_extractor.process(img, superpxs)
graph = rag_graph_builder.process(superpxs, feats)
```

Input Image | Cell Graph | Tissue Graph

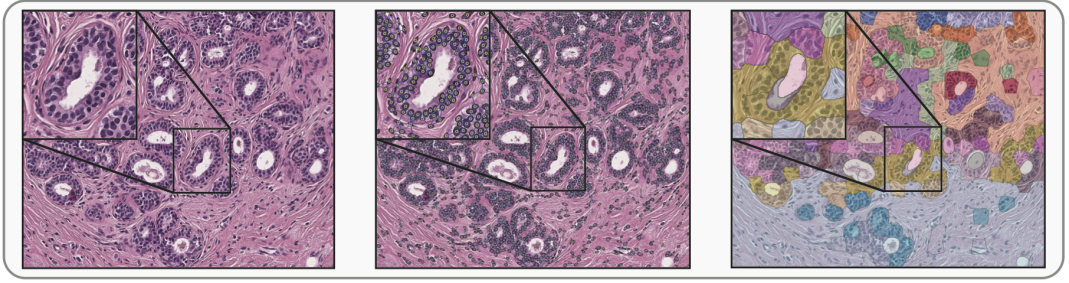


Figure B.3 – Implementation of cell-graph (left) and tissue-graph (right) generation using the graph builders in HISTOCARTOGRAPHY.

B.3.2 Graph machine learning module

HISTOCARTOGRAPHY includes a set of DL models, based on a GNN backbone to learn from graph-structured tissue representations. It includes two state-of-the-art GNN layers, *i.e.*, GIN (Xu et al., 2019b) and PNA (Corso et al., 2020). PNA proves to outperform GIN provided more computational resources (Dwivedi et al., 2020) (see Chapter 2). HISTOCARTOGRAPHY defines cell- and tissue-graph models, which are GNN-based abstractions to learn from biological entity-graphs. They offer efficient (Pati et al., 2021a), scalable (Anklin et al., 2021; Jaume et al., 2021c) and explainable (Jaume et al., 2020, 2021b) approaches to analyze histology images. Further, the library includes models to jointly represent and learn from cell- and tissue-

graphs (see Chapter 5). The models in HISTOCARTOGRAPHY are organized such that they can be adapted to various GNN backbones, tasks (*e.g.*, regression, clustering, classification, segmentation), organs, and entity-types. These models provide blueprints to accelerate the development of graph-based models in computational pathology. All the graph modules are implemented using DGL (Wang et al., 2019a), a state-of-the-art library for GNNs built around PyTorch. Figure B.4 presents the syntax to declare and run a cell- and tissue-graph model. All the model parameters, *e.g.*, GNN type, number of GNN layers, can be adapted and fine-tuned using a configuration file.

Cell Graph Model

```python
import yaml
from dgl.data.utils import (
  load_graphs
)
from histocartography.ml import (
  CellGraphModel
)

# load model configurations
cfg = yaml.safe_load(open('cg_cfg.yml', 'r'))

# define cell graph model
model = CellGraphModel(
    gnn_params=cfg['gnn_params'],
    classification_params=cfg['cls_params'],
    node_dim=512,
    num_classes=3
)

# load cell graph
cg, _ = load_graphs('cg.bin')

# forward pass
logits = model(cg)
```

Tissue Graph Model

```python
import yaml
from dgl.data.utils import (
  load_graphs
)
from histocartography.ml import (
  TissueGraphModel
)

# load model configurations
cfg = yaml.safe_load(open('tg_cfg.yml', 'r'))

# define tissue graph model
model = TissueGraphModel(
    gnn_params=cfg['gnn_params'],
    classification_params=cfg['cls_params'],
    node_dim=512,
    num_classes=3
)

# load tissue graph
tg, _ = load_graphs('tg.bin')

# forward pass
logits = model(tg)
```

Figure B.4 – Implementation of the cell- (left) and tissue- graph (right) model by using the ML modules in the HISTOCARTOGRAPHY API

.

B.3.3 Explainability module

HISTOCARTOGRAPHY includes four post-hoc feature attribution graph explainers, that can generate node-level saliency maps to highlight the node-wise contribution towards an output task. Namely, the library includes two gradient-based explainers (GRAPHGRAD-CAM (Selvaraju et al., 2017; Pope et al., 2019) and GRAPHGRAD-CAM++ (Chattopadhay et al., 2018; Jaume et al., 2021b)), a node pruning-based explainer (GNNEXPLAINER (Ying et al., 2019)), and a layer-wise relevance propagation explainer (GRAPHLRP (Schwarzenberg et al., 2019)). The saliency map can be visualized by overlaying the node importances on the input image (see Figure B.6). Alternatively, entities with high importances can be extracted and studied independently to assess their relevance (see Chapter 6). Figure B.5 shows code snippets to use the graph explainability modules. All explainers follow a similar syntax with the same input and output types, making implementation and integration straightforward.

Cell Graph Explainer Code

```python
from histocartography.interpretability import (
  GraphGradCAMExplainer,
  GraphGradCAMPPExplainer,
  GraphPruningExplainer,
  GraphLRPExplainer
)

# load pretrained model
model = CellGraphModel(config['gnn_params'], config['cls_params'], 512, pretrained=True)

# load cell graph
graph, _ = load_graphs('291_dcis_18.bin')

# define graph explainers
grad_cam_explainer = GraphGradCAMExplainer(model=model)
grad_campp_explainer = GraphGradCAMPPExplainer(model=model)
gnn_explainer = GraphPruningExplainer(model=model)
graph_lrp_explainer = GraphLRPExplainer(model=model)

# explain cell graph
grad_cam_scores, _ = grad_cam_explainer.process(graph)
grad_campp_scores, _ = grad_campp_explainer.process(graph)
gnn_explainer_scores, _ = gnn_explainer.process(graph)
graph_lrp_scores, _ = graph_lrp_explainer.process(graph)
```

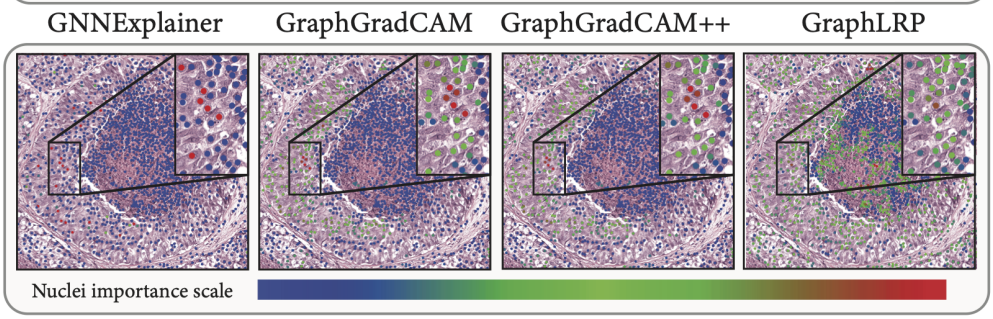


Figure B.5 – Implementation of graph explainers in HistoCartography. The most important nodes are marked in red and the least important ones in blue.

B.3.4 Pipeline runner

To facilitate an easy-to-use and human-readable development, HistoCartography includes a pipeline runner. It allows to define a series of pipeline steps along with loading and saving utilities to reduce boilerplate code.

B.4 Benchmarking HistoCartography

We benchmark HistoCartography in terms of run-time and performance for various histopathology tasks, *i.e.*, stain normalization, tissue detection, tumor classification and segmentation etc., on images of varying dimensions. The CPU and GPU compatible modules are assessed on a single-core POWER9 processor and a NVIDIA P100 GPU, respectively.

B.4.1 Computational time

Analyzing the computational time for processing a histology image is imperative for deploying applications in real-life settings. We thoroughly analyze the run-time of HistoCartography modules on a set of RoIs and WSIs. The analyzes are presented in Table B.2. The preprocessing

modules are observed to be the most time-consuming. For instance, Vahadane stain normalization can take up to 3 minutes to process a 11′000 × 11′000 image, whereas Macenko method is 2× faster for competitive result. The implementations are computationally similar to HistoLab and StainTools, and scale linearly w.r.to image size. The cell- and tissue-graph construction take 2.5 and 4.1 seconds, respectively, for a 1000 × 1000 image with the following parameters. Nuclei detection is performed on patches of size 256 × 256 with an overlap of 164 pixels. Nuclei features are extracted from 72 × 72 patches centered around the nuclei, that are resized to 224 × 224 and processed by ResNet34 pretrained on ImageNet (Deng et al., 2009a). Finally, thresholded k-NN topology is built with $k = 5$ and a threshold distance of 50 pixels. For the tissue-graph, SLIC is used to extract 400 superpixels per image, that are subsequently merged to provide the tissue components. Tissue features are also extracted using ResNet34 with 144 × 144 size patches that are resized to 224 × 224. The graph buildings can be further optimized as per the task by downsampling the input image, reducing the patch overlap, or by using a lighter feature extractor. For extracting the feature cube representation, we process patches of size 144 × 144 resized to 224 × 224 w/o overlap by pretrained ResNet34.

TRoIs are processed using a cell- and tissue-graph model, and the hierarchical cell-to-tissue graph model as introduced in Chapter 5. They consist of three PNA layers with 64 hidden units followed by an additional 2-layer MLP with 128 hidden units for classification. WSIs are processed using WholeSIGHT (see Chapter 7), which contains six GIN layers with 64 hidden units followed by a 2-layer MLP with 128 hidden units. The models process in near real-time irrespective of the increment in the graph size. The graph explainers are based on GNNs with 3 GIN layers, each having a 2-layer MLP with 32 hidden units, and a 2-layer MLP head. GNNExplainer is the slowest among all as it requires to optimize a mask to explain each image.

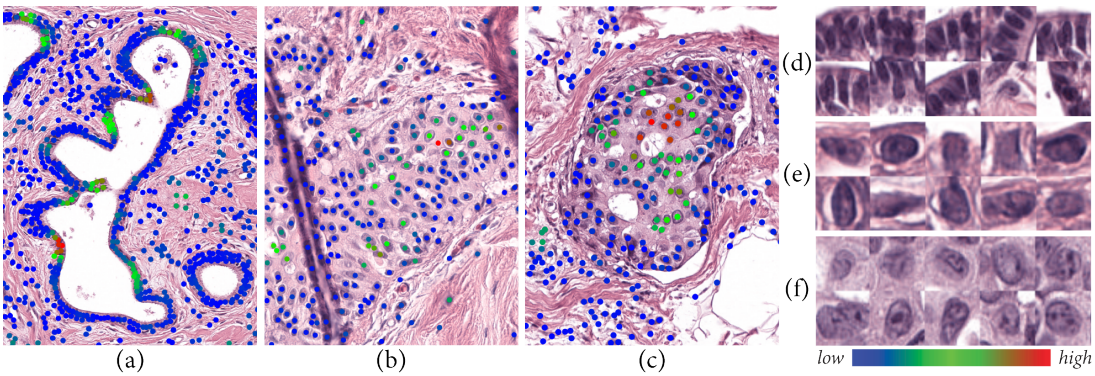


Figure B.6 – Qualitative explanations of sample breast RoI: (a) Benign, (b) ADH, (c) DCIS. (d, e, f) highlight the ten most important nuclei for the respective samples.

B.4.2 Performance benchmark

Table B.3 benchmarks the performance of HistoCartography for classification and segmentation tasks. Classification is performed on BRACS (Pati et al., 2021a) and BACH (Aresta

et al., 2019) datasets to characterize breast tumors using a cell-graph model, a tissue-graph model, and HACT-Net. The performance is measured by weighted-F1 score. Segmentation is performed using WHOLESIGHT to delineate Gleason patterns in prostate cancer images from UZH (Zhong et al., 2017) and SICAPv2 (Silva-Rodríguez et al., 2020), and the performance is measured by average Dice score. We evaluate on various image types, *i.e.*, tumor RoIs, tissue microarrays, and whole-slides, to highlight the scalability of entity-graphs in HISTOCARTOGRAPHY to arbitrary image dimensions.

B.4.3 Qualitative explanations

Figure B.6 presents the outcome of GRAPHGRADCAM module in HISTOCARTOGRAPHY to interpret a cell-graph model. This module renders per-image explanations in terms of node-level saliency maps by applying post-hoc feature attribution methods on trained cell-graph model. Further, the cell-graph model can be interpreted by characterizing the highlighted important nuclei per-image, as shown in Figure B.6.

		Modality	Tumor RoI			WSI		
		Size	1000^2	2500^2	5000^2	5000^2	7500^2	11000^2
Preprocessing	Standard	Vahadane Normalization	1.77	6.46	29.03	30.67	68.27	186.10
		Macenko Normalization	0.80	2.86	11.19	15.98	32.37	81.72
		Tissue Mast Detection	-	-	-	1.04	2.11	8.09
		Feature Cube Extraction	0.24	1.61	5.92	6.27	11.97	29.79
	CG	Nuclei Detection	3.03	12.93	47.66	-	-	-
		Nuclei Concept Extraction	2.95	6.52	27.94	-	-	-
		Deep Nuclei Feature Extraction	0.10	0.30	1.28	-	-	-
		k-NN Graph Building	0.06	0.20	1.35	-	-	-
	TG	Super-pixel Detection	3.32	17.84	68.99	31.50	68.99	183.54
		Deep Tissue Feature Extraction	0.56	2.99	8.40	4.17	9.96	20.54
		RAG Graph Building	0.12	2.04	25.6	6.33	19.98	85.73
ML		Cell-Graph Model	0.028	0.033	0.040	-	-	-
		Tissue-Graph Model	0.011	0.015	0.026	0.039	0.056	0.069
		HACT Model	0.034	0.041	0.057	-	-	-
Explainers	CG	GNNEXPLAINER	12.00	13.09	35.33	-	-	-
		GRAPHGRAD-CAM	0.011	0.022	0.035	-	-	-
		GRAPHGRAD-CAM++	0.011	0.023	0.035	-	-	-
		GRAPHLRP	0.020	0.024	0.90	-	-	-
	TG	GNNEXPLAINER	11.23	11.28	11.38	-	-	-
		GRAPHGRAD-CAM	0.011	0.012	0.018	0.025	0.030	0.033
		GRAPHGRAD-CAM++	0.011	0.013	0.018	0.026	0.030	0.033
		GRAPHLRP	0.011	0.014	0.016	0.079	0.085	0.089

Table B.2 – Reported time to run HISTOCARTOGRAPHY core functionalities. CPU-only experiments were run on a single-core POWER8 processor, and GPU-compatible experiments were run on an NVIDIA P100 GPU. Time is reported in seconds.

Task	Dataset	Model	Image Type	Avg. #pixels	#classes	Avg. Dice	Weighted F1
Classification	BRACS	CG-GNN	TRoI	3.9×10^6 (40×)	7	-	55.9 ± 1.0
	BRACS	TG-GNN	TRoI	3.9×10^6 (40×)	7	-	56.6 ± 1.3
	BRACS	HACT-Net	TRoI	3.9×10^6 (40×)	7	-	61.5 ± 0.9
	BACH	HACT-Net	TRoI	3.1×10^6 (20×)	4	-	90.7 ± 0.5
	SICAPv2	SEGGINI	WSI	121×10^6 (10×)	6	-	62.0 ± 3.6
	UZH	SEGGINI	TMA	9.6×10^6 (40×)	6	-	56.8 ± 1.7
Seg.	SICAPv2	SEGGINI	WSI	121×10^6 (10×)	4	44.3 ± 2.0	-
	UZH	SEGGINI	TMA	9.6×10^6 (40×)	4	66.0 ± 3.1	-

Table B.3 – Benchmarking HISTOCARTOGRAPHY for classification and segmentation (in %).

C Qualitative Assessment of Graph Explainers

Figure C.1 and Figure C.2 present cell-graph explanations produced by GNNEXPLAINER, GRAPHGRAD-CAM, GRAPHGRAD-CAM++ and GRAPHLRP for benign, atypical and malignant breast tumors. It can be observed that GNNEXPLAINER learns to binarize the explanations, thereby producing the most compact explanations by retaining the most important nuclei set of nuclei with high importance. However, GRAPHGRAD-CAM and GRAPHGRAD-CAM++ produce explanations with more distributed nuclei importance than GNNEXPLAINER. GRAPHLRP produces the largest explanations by retaining most of the nuclei in the cell-graphs.

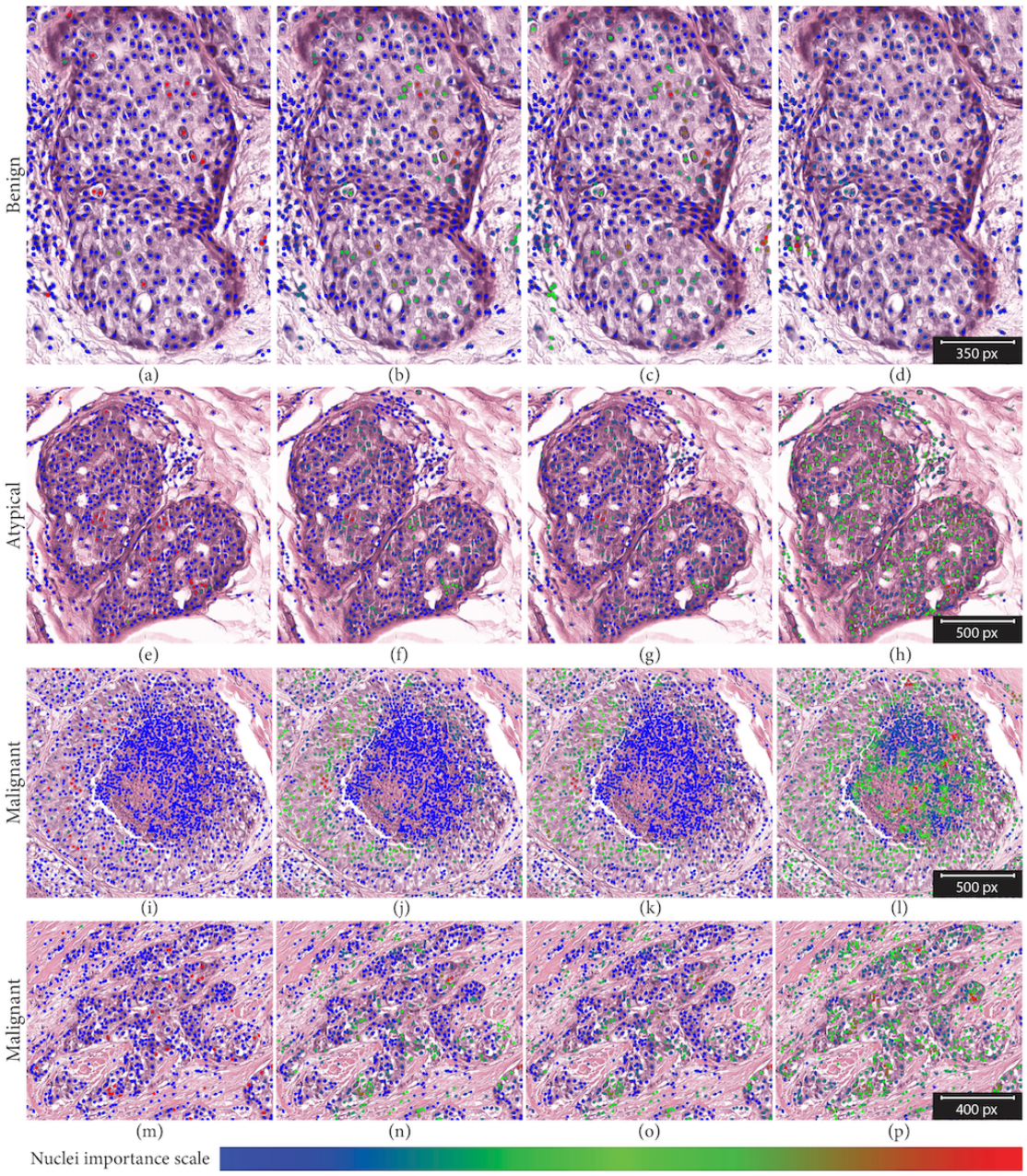


Figure C.1 – Qualitative results. The rows represent breast cancer subtypes, and columns represent graph explainers, *i.e.*, GNNEXPLAINER, GRAPHGRAD-CAM, GRAPHGRAD-CAM++, and GRAPHLRP. Nuclei level importance ranges from blue (the least important) to red (the highest important).

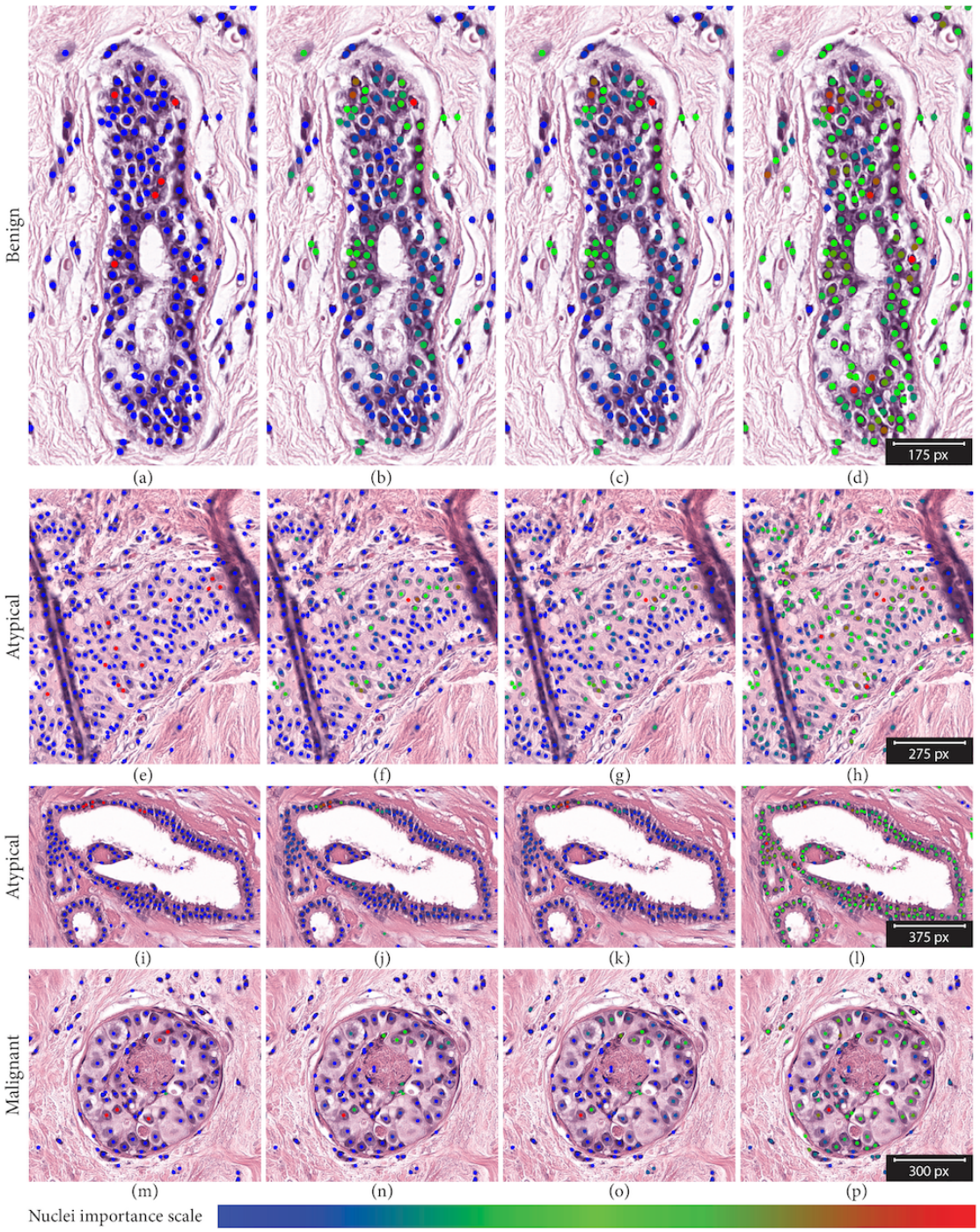


Figure C.2 – Qualitative results. The rows represent breast cancer subtypes, and columns represent graph explainers, *i.e.*, GNNEXPLAINER, GRAPHGRAD-CAM, GRAPHGRAD-CAM++, and GRAPHLRP. Nuclei level importance ranges from blue (the least important) to red (the highest important).

D Extension of Weakly Supervised Learning for Joint Whole-Slide Segmentation and Classification

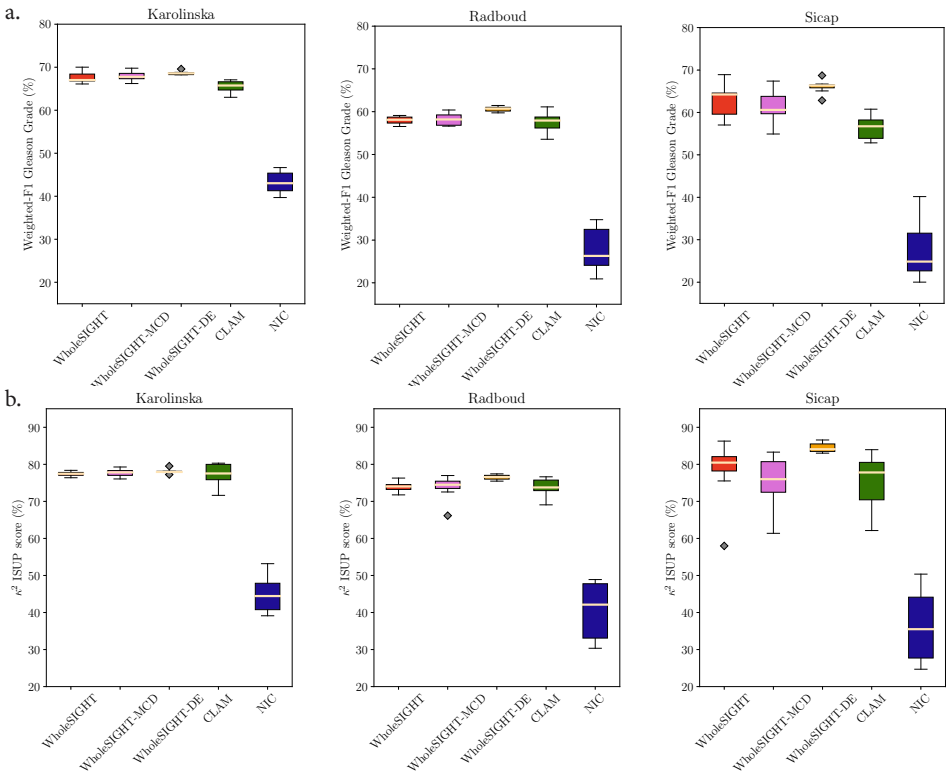


Figure D.1 – (a) Gleason grade classification measured with weighted-F1 scores for WHOLESIGHT, WHOLESIGHT-MCD, WHOLESIGHT-DE, CLAM, and NIC methods (higher is better). (b) Quadratic Cohen's Kappa scores (κ^2) of ISUP classification obtained for WHOLESIGHT, WHOLESIGHT-MCD, WHOLESIGHT-DE, CLAM, and NIC (higher is better).

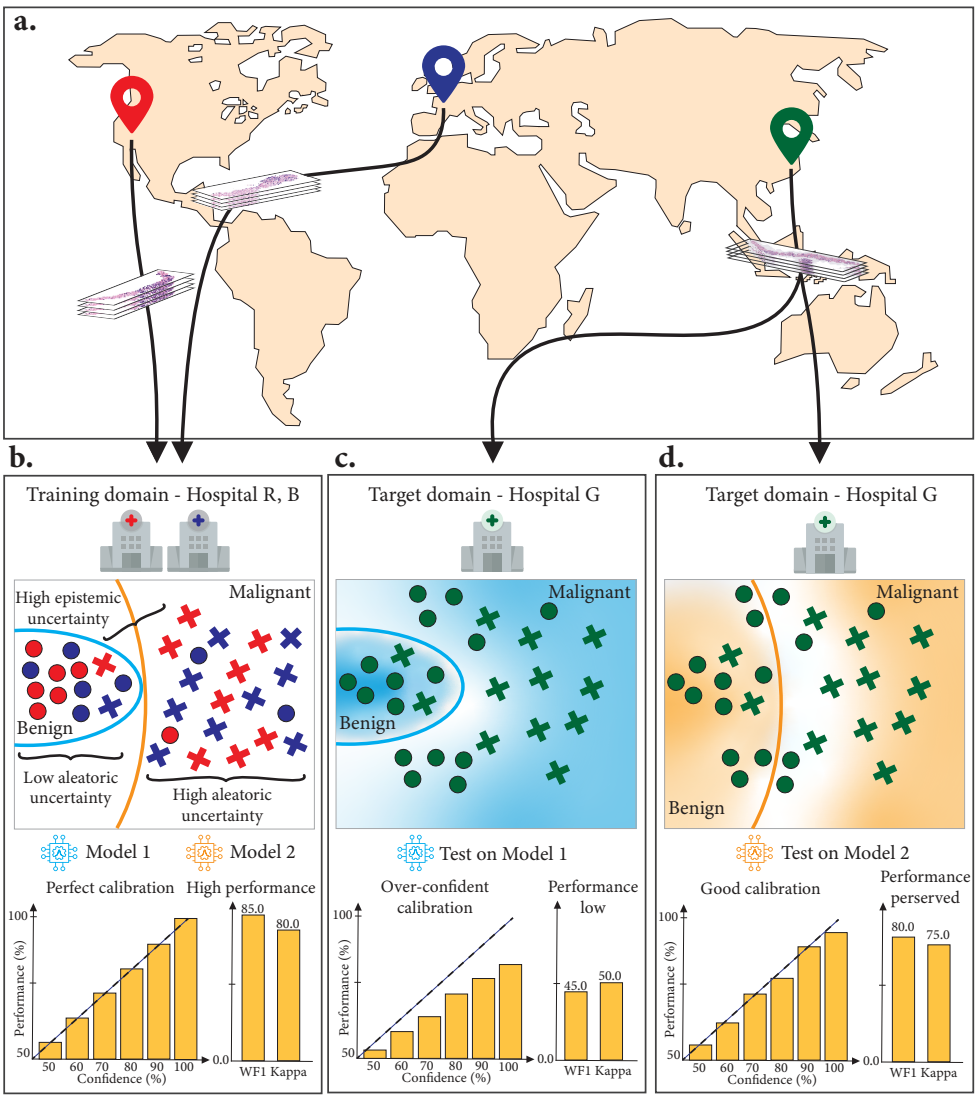


Figure D.2 – (a) Each hospital R, B, G (marked in red, blue and green) represents a different dataset. Due to varying slide acquisition protocols and demographics variability, local hospital-level biases are introduced in the data. (b) To address this variability, a dataset composed of samples from different hospitals (R and B in this scenario) is created. A DL system is trained until satisfying testing performance is reached on hospital R and B. In this toy example, the benign class has lower *aleatoric uncertainty* than the malignant one. Even if models perform similarly, the learned decision boundaries can differ (see orange and light blue models), which is referred to as *epistemic uncertainty*. Good models should generalize as well as possible to unseen data while providing accurate confidence estimates in case of domain shifts. In (c) and (d), we study model generalization on hospital G. (c) Model 1 is showing poor generalization capabilities and calibration, making it hard to detect the domain shift. (d) Model 2, with smoother decision boundaries, results in both better performance and confidence predictions, where misclassified samples are also not confident (*i.e.*, they lie close to the decision boundary). Overall, Model 2 leads to better calibration than Model 1 by offering more robust predictions.

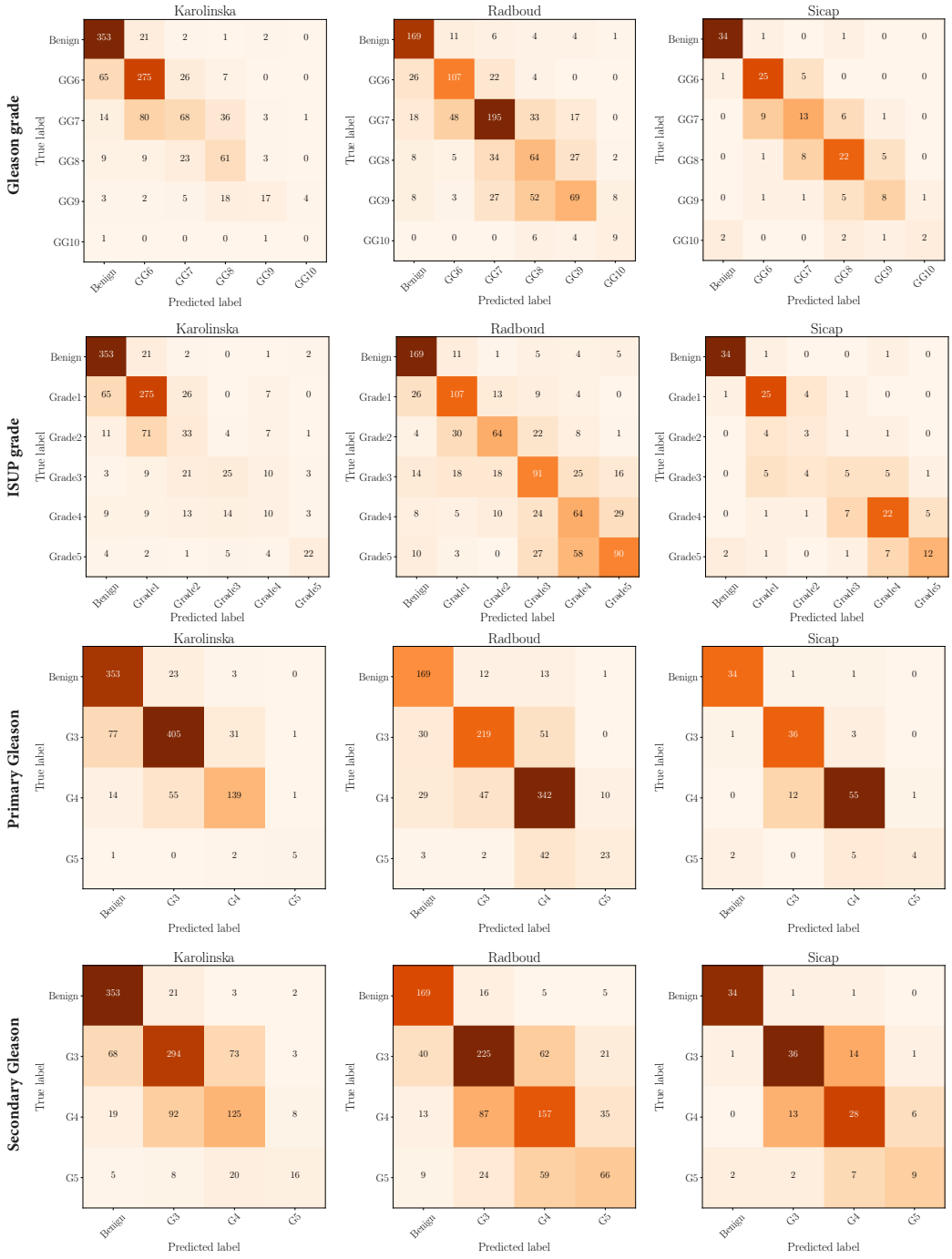


Figure D.3 – Gleason grade, ISUP grade, primary Gleason score classification, and secondary Gleason score classification confusion matrices obtained for WHOLESIGHT-DE on the Karolinska, Radboud, and Sicap datasets.

Bibliography

Achanta, R. et al. (2012). Slic superpixels compared to state-of-the-art superpixel methods. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 34, pages 2274–2282.

Adnan, M. et al. (2020). Representation learning of histopathology images using graph neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 4254–4261.

Ahmedt, D. et al. (2021). A survey on graph-based deep learning for computational histopathology. In *arXiv:2107.00272*.

Allison, K., Rendi, M., Peacock, S., Morgan, T., Elmore, J., and Weaver, D. (2016). Histologic Features associated with Diagnostic Agreement in Atypical Ductal Hyperplasia of the Breast: Illustrative Cases from the B-Path Study. *Histopathology*, 69(6):1028–1046.

Amin, M. et al. (2014). The critical role of the pathologist in determining eligibility for active surveillance as a management option in patients with prostate cancer: consensus statement with recommendations supported by the college of american pathologists, international society of urological pathology, association of directors of anatomic and surgical pathology, the new zealand society of pathologists, and the prostate cancer foundation. *Arch Pathol Lab Med*.

Anand, D. et al. (2020). Histographs: graphs in histopathology. In *SPIE Medical Imaging 2020: Digital Pathology*, volume 11320, page 113200O.

Andriluka, M. et al. (2014). 2d human pose estimation: New benchmark and state of the art analysis. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3686–3693.

Anklin, V. et al. (2021). Learning whole-slide segmentation from inexact and incomplete labels using tissue graphs. In *Medical Image Computing and Computer Assisted Intervention (MICCAI)*.

Araujo, T. et al. (2005). Classification of breast cancer histology images using convolutional neural networks. In *PloS one*, volume 12.

Arbitrio, E. et al. (2020). histolab.

Aresta, G. et al. (2019). Bach: Grand challenge on breast cancer histology images. In *Medical Image Analysis*, volume 56, pages 122–139.

Arrieta, A. et al. (2020). Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges toward Responsible AI. *Information Fusion*, 58:82–115.

Aubreville, M. et al. (2020). Inter-species, inter-tissue domain adaptation for mitotic figure assessment: Learning new tricks from old dogs. In *Bildverarbeitung für die Medizin 2020*.

Aubreville, M. et al. (2021). Quantifying the scanner-induced domain gap in mitosis detection. In *Medical Imaging with Deep Learning (MIDL)*.

Aygunes, B. et al. (2020). Graph convolutional networks for region of interest classification in breast histopathology. In *SPIE Medical Imaging 2020: Digital Pathology*, volume 11320, page 113200K.

Bach, S. et al. (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS ONE*, 10(7).

Baldassarre, F. and Azizpour, H. (2019). Explainability Techniques for Graph Convolutional Networks. *International Conference on Machine Learning Workshops*.

Bankhead, P. et al. (2017). Qupath: Open source software for digital pathology image analysis. In *Scientific reports*, volume 7, pages 1–7.

Bankhead, P. et al. (2021). Qupath.

Bardou, D. et al. (2018). Classification of breast cancer based on histology images using convolutional neural networks. In *IEEE Access*, volume 6, pages 24680–24693.

Battaglia, P. W. et al. (2018). Relational inductive biases, deep learning, and graph networks. *arXiv:1806.01261*.

Beck, D. et al. (2018). Graph-to-sequence learning using gated graph neural networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 273—-283.

Beezley, J. et al. (2021). Histomicstk.

Bejnordi, B. et al. (2015). A multi-scale superpixel classification approach to the detection of regions of interest in whole slide histopathology images. In *SPIE 9420, Medical Imaging 2015: Digital Pathology*, volume 94200H.

Bejnordi, B. et al. (2017). Context-aware stacked convolutional neural networks for classification of breast carcinomas in whole-slide histopathology images. In *Journal of Medical Imaging*, volume 4.

Bejnordi, B. et al. (2019). Diagnostic assessment of deep learning algorithms for detection of lymph node metastases in women with breast cancer. In *JAMA*, volume 318, page 2199–2210.

Bera, K. et al. (2019). Artificial intelligence in digital pathology — new tools for diagnosis and precision oncology. *Nature Reviews Clinical Oncology*, 16:703–715.

Binder, A. et al. (2018). Towards computational fluorescence microscopy: Machine learning-based integrated prediction of morphological and molecular tumor profiles. *arXiv:1805.11178*.

Binder, T. et al. (2019). Multi-organ gland segmentation using deep learning. In *Frontiers in Medicine*.

Boyce, B. et al. (2017). An update on the validation of whole slide imaging systems following fda approval of a system for a routine pathology diagnostic service in the united states. In *Chinese Journal of Cancer Research*.

Brancati, N. et al. (2018). Multi-classification of breast cancer histology images by using a fine-tuning strategy. In *International Conference Image Analysis and Recognition*, pages 771–778.

Bronstein, M. et al. (2017). Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*.

Bruna, J. et al. (2014). Spectral Networks and Deep Locally Connected Networks on Graphs. *International Conference on Learning Representations (ICLR)*.

Bruno, K. et al. (2017). Looking under the hood Deep neural network visualization to interpret whole slide Image analysis outcomes for colorectal polyps. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*.

Bulten, W. et al. (2020). Automated deep-learning system for Gleason grading of prostate cancer using biopsies: a diagnostic study. In *Lancet Oncology*.

Bulten, W. et al. (2021). Artificial intelligence assistance significantly improves gleason grading of prostate biopsies by pathologists. In *Modern Pathology*.

Byfield, P. et al. (2019). Staintools.

Byfield, P. et al. (2020). Syntax.

Cai, J. et al. (1992). An optimal lower bound on the number of variables for graph identification. *Combinatorica*.

Campanella, G. et al. (2019). Clinical-grade computational pathology using weakly supervised deep learning on whole slide images. In *Nature Medicine*, volume 25, page 1301–1309.

Chan, L. et al. (2019). Histosegnet: Semantic segmentation of histological tissue type in whole slide images. In *IEEE ICCV*, pages 10661–10670.

Chan, L. et al. (2021). A comprehensive analysis of weakly-supervised semantic segmentation in different image domains. In *IJCV*, volume 129, pages 361–384.

Chattopadhay, A. et al. (2018). Grad-CAM++: Generalized gradient-based visual explanations for deep convolutional networks. In *IEEE Winter Conference on Applications of Computer Vision*, volume 2018-Janua, pages 839–847.

Chen, N. et al. (2008). The evolving gleason grading system. In *Chinese Journal of Cancer Research*.

Chen, R. et al. (2020). Pathomic fusion: An integrated framework for fusing histopathology and genomic features for cancer diagnosis and prognosis. In *IEEE Transactions on Medical Imaging*.

Cheng, L. et al. (2007). Percentage of gleason pattern 4 and 5 predicts survival after radical prostatectomy. *Cancer*.

Chennamsetty, S. et al. (2018). Classification of breast cancer histology image using ensemble of pre-trained neural networks. In *International Conference Image Analysis and Recognition*, pages 804–811.

Choy, B. et al. (2016). Prognostic significance of percentage and architectural types of contemporary gleason pattern 4 prostate cancer in radical prostatectomy. *Am J Surg Pathol*.

Cid, S. et al. (2018). Prognostic influence of tumor stroma on breast cancer subtypes. In *Clinical Breast Cancer*, volume 18, pages 123–133.

Corso, G. et al. (2020). Principal neighbourhood aggregation for graph nets. In *Neural Information Processing Systems (NeurIPS)*.

Cybenkot, G. (1989). Mathematics of control, signals, and systems approximation by superpositions of a sigmoidal function. *Math. Control Signals Systems*, 2:303–314.

Dai, H. et al. (2016). Discriminative embeddings of latent variable models for structured data. *International Conference on Machine Learning (ICML)*.

Danilevsky, M. et al. (2020). A survey of the state of explainable ai for natural language processing. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 447–459, Suzhou, China. Association for Computational Linguistics.

De Vries, G. et al. (2015). Substructure counting graph kernels for machine learning from rdf data. *Journal of Web Semantics*.

Debnath, A. K. et al. (1991). Structure-Activity Relationship of Mutagenic Aromatic and Heteroaromatic Nitro Componds. Correlation with Modelcular Orbital Energies and Hydrophobicity. *J. Med. Chem*, 34:786–797.

Defferrard, M. et al. (2016). Convolutional neural networks on graphs with fast localized spectral filtering. In *Neural Information Processing Systems (NeurIPS)*, pages 3844–3852.

DeGroot, M. H. et al. (1983). The comparison and evaluation of forecasters. In *Journal of the Royal Statistical Society: Series D (The Statistician)*.

Dehmamy, N. et al. (2019). Understanding the representation power of graph neural networks in learning graph topology. In *Neural Information Processing Systems (NeurIPS)*, pages 15413–15423.

Demir, C. et al. (2004). The cell graphs of cancer. In *Bioinformatics*, volume 20, pages 145–151.

Deng, J. et al. (2009a). Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255.

Deng, J. et al. (2009b). ImageNet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255.

Deng, S. et al. (2020). Deep learning in digital pathology image analysis: A survey. In *Frontiers of Medicine*.

Dhurandhar, A. et al. (2017). A Formal Framework to Characterize Interpretability of Procedures. In *International Conference on Machine Learning*.

Ding, K. et al. (2020). Feature-enhanced graph networks for genetic mutational prediction using histopathological images in colon cancer. In *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*.

Doshi-Velez, F. and Kim, B. (2017). Towards a rigorous science of interpretable machine learning. In *arXiv:1702.08608*.

Duggan, M. et al. (2021). Surveillance, epidemiology, and end results (seer) 18 registries. *National Cancer Institute*.

Dwivedi, V. et al. (2020). Benchmarking graph neural networks. In *arXiv:2003.00982*.

Elmore, J. et al. (2015). Diagnostic concordance among pathologists interpreting breast biopsy specimens. In *JAMA*, volume 313, pages 1122–1132.

Epstein, J. et al. (2005). The 2005 international society of urological pathology (isup) consensus conference on gleason grading of prostatic carcinoma. *The American journal of surgical pathology*.

Epstein, J. et al. (2014). The 2014 international society of urological pathology (isup) consensus conference on gleason grading of prostatic carcinoma: Definition of grading patterns and proposal for a new grading system. *The American journal of surgical pathology*.

Faryna, K. et al. (2021). Tailoring automated data augmentation to h&e-stained histopathology. In *Medical Imaging with Deep Learning (MIDL)*.

Fort, S. et al. (2019). Deep ensembles: A loss landscape perspective. *Advances in Neural Information Processing Systems (NeurIPS)*.

Francis, K. and Palsson, B. (1997). Effective intercellular communication distances are determined by the relative time constants for cyto/chemokine secretion and diffusion. *Proceedings of the National Academy of Sciences*, 94(23):12258–12262.

Gal, Y. and Ghahramani, Z. (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of The 33rd International Conference on Machine Learning*.

Gamble, P. et al. (2021). Determining breast cancer biomarker status and associated morphological features using deep learning. In *Nature Communications*.

Gamper, J. et al. (2019). Pannuke dataset extension, insights and baselines. pages 11–19.

Ganin, Y. et al. (2016). Domain-adversarial training of neural networks. In *The Journal of Machine Learning Research*.

Garciá-Arteaga, J. et al. (2017). A lymphocyte spatial distribution graph-based method for automated classification of recurrence risk on lung cancer images. In *International Symposium on Medical Information Processing and Analysis*, volume 10956, page 109560H.

Gilbert, B. et al. (2020). Openslide.

Gilmer, J. et al. (2017). Neural message passing for quantum chemistry. In *International Conference on Machine Learning (ICML)*, volume 70, pages 1263–1272.

Gomariz, A. et al. (2021). Probabilistic spatial analysis in quantitative microscopy with uncertainty-aware cell detection using deep bayesian regression of density maps. In *arXiv:2102.11865*.

Gomes, D. et al. (2014). Inter-observer variability between general pathologists and a specialist in breast pathology in the diagnosis of lobular neoplasia, columnar cell lesions, atypical ductal hyperplasia and ductal carcinoma in situ of the breast. In *Diagnostic Pathology*, volume 9.

Graham, S. et al. (2019a). Hover-net: Simultaneous segmentation and classification of nuclei in multi-tissue histology images. In *Medical Image Analysis*, volume 58.

Graham, S. et al. (2019b). Mild-net: Minimal information loss dilated network for gland instance segmentation in colon histology images. In *Medical Image Analysis*, volume 52, pages 199–211.

Graziani, M. et al. (2020). Concept attribution: Explaining CNN decisions to physicians. In *Computers in Biology and Medicine*, volume 123.

Grohe, M. et al. (2017). Color Refinement and its Applications. In *An Introduction to Lifted Probabilistic Inference*.

Gunduz, C. et al. (2004). The cell graphs of cancer. *Bioinformatics*, 20(1):145–151.

Guo, H. et al. (2019). On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*.

Gustafsson, F. K. et al. (2019). Evaluating scalable bayesian deep learning methods for robust computer vision. In *Conference on Computer Vision and Pattern Recognition Workshops (CVPR-W)*.

Hagele, M. et al. (2020). Resolving challenges in deep learning-based analyses of histopathological images using explanation methods. *Nature Scientific Reports*, 10.

Hamilton, W. et al. (2017). Inductive representation learning on large graphs. In *Neural Information Processing Systems (NeurIPS)*, pages 1024–1034.

Hamilton, W. L. et al. (2020). Graph Representation Learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*.

Haralick, R. et al. (1973). Texural features for image classification. *IEEE transaction on systems, man and cybernatics*, 3(6):610–621.

Harris, R. P. et al. (2020). Array programming with NumPy. *Nature*, 585(7825):357–362.

He, H. and Ma, Y. (2013). *Imbalanced Learning: Foundations, Algorithms, and Applications*. Wiley-IEEE Press, 1st edition.

He, K. et al. (2016). Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778.

He, K. et al. (2017). Mask R-CNN. *Proceedings of the IEEE International Conference on Computer Vision*, 2017-Octob:2980–2988.

Helma, C. et al. (2003). The Predictive Toxicology Challenge 2000-2001. *Bioinformatics (Oxford, England)*, 19(1):1179–82.

Hinton, G. et al. (2015). Distilling the Knowledge in a Neural Network. In *Advances in neural information processing systems (NeurIPS)*.

Ho, D. et al. (2021). Deep multi-magnification networks for multi-class breast cancer image segmentation. In *Computerized Medical Imaging and Graphics*, volume 88, page 101866.

Hoffman, R. et al. (2018). Metrics for explainable AI: Challenges and prospects. In *arXiv:1812.04608*.

Holzinger, A. et al. (2017). Towards the Augmented Pathologist: Challenges of Explainable-AI in Digital Pathology. In *arXiv:1712.06657*.

Hou, L. et al. (2016). Patch-based convolutional neural network for whole slide tissue image classification. In *IEEE CVPR*, page 2424–2433.

Huang, C. et al. (2014). Gleason score 3+4=7 prostate cancer with minimal quantity of gleason pattern 4 on needle biopsy is associated with low-risk tumor in radical prostatectomy specimen. *Am J Surg Pathol*.

Huang, Q. et al. (2021). Graphlime: Local interpretable model explanations for graphneural networks. In *arXiv preprint arXiv:2001.06216*.

Ibrahim, A. et al. (2020). Artificial intelligence in digital breast pathology: Techniques and applications. In *The Breast*, volume 49, pages 267–273.

Jampani, V. et al. (2018). Superpixel sampling networks. In *European Conference on Computer Vision (ECCV)*.

Jaume, G. et al. (2019). edgnn: a simple and powerful gnn for directed labeled graphs. In *International Conference on Learning Representations (ICLR) Workshop on Representation Learning on Graphs and Manifolds*.

Jaume, G. et al. (2020). Towards explainable graph representations in digital pathology. In *International Conference on Machine Learning (ICML), Workshop on Computational Biology*.

Jaume, G. et al. (2021a). Histocartography: A toolkit for graph analytics in digital pathology. In *Third MICCAI workshop on Computational Pathology (MICCAI-W)*.

Jaume, G. et al. (2021b). Quantifying explainers of graph neural networks in computational pathology. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jaume, G. et al. (2021c). Weakly supervised learning for joint whole-slide segmentation and classification in prostate cancer. *Preprint*.

Javed, S. et al. (2020). Cellular community detection for tissue phenotyping in colorectal cancer histology images. In *Medical Image Analysis*, volume 63.

Jia, Z. et al. (2017). Constrained deep weak supervision for histopathology image segmentation. In *IEEE Transactions on Medical Imaging*, volume 36, pages 2376–2388.

Kashyap, A. et al. (2018). Role of Nuclear Morphometry in Breast Cancer and its Correlation with Cytomorphological Grading of Breast Cancer: A Study of 64 Cases. *Journal of Cytology*, 35(1):41–45.

Kendall, A. and Yarin, G. (2017). What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in Neural Information Processing Systems*.

Kenton, M. C. et al. (2017). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of NAACL-HLT*.

Kiefer, S. et al. (2020). Power and limits of the weisfeiler-leman algorithm. *Dissertation, RWTH Aachen University*.

Kim, B. et al. (2018). Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV). *International Conference on Machine Learning*, page 2673–2682.

Kindermans, P. et al. (2015). PatternNet and PatternLRP - improving the interpretability of neural networks. *arXiv:1705.05598*.

Kingma, D. and Ba, J. (2015). Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*.

Kipf, T. and Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations (ICLR)*.

Komura, D. and Ishikawa, S. (2018). Machine learning methods for histopathological image analysis. In *Computational and Structural Biotechnology Journal*, pages 34–42.

Krause, J. et al. (2013). 3d object representations for fine-grained categorization. In *IEEE International Conference on Computer Vision Workshops*, pages 554–561.

Kriege, N. et al. (2012). Subgraph Matching Kernels for Attributed Graphs. In *International Conference on Machine Learning (ICML)*.

Krizhevsky, A. et al. (2012). ImageNet Classification with Deep Convolutional Neural Networks. *Advances in Neural Information Processing Systems 25 (NIPS2012)*, pages 1–9.

Kumar, A. et al. (2018). Trainable calibration measures for neural networks from kernel mean embeddings. In *International Conference on Machine Learning (ICML)*.

Kumar, N. et al. (2017). A dataset and a technique for generalized nuclear segmentation for computational pathology. In *IEEE Transactions on Medical Imaging*, volume 36, pages 1550–1560.

Lakshminarayanan, B. et al. (2017). Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*.

Lerwill, M. F. (2008). Flat epithelial atypia of the breast. In *Archives of pathology & laboratory medicine*, volume 132, pages 615–21.

Levy, J. et al. (2021). Topological feature extraction and visualization of whole slide images using graph neural networks. In *Proceedings of the Pac. Symposium on Biocomputing (PSB)*.

Li, R. et al. (2018a). Graph cnn for survival analysis on whole slide pathological images. In *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*.

Li, Y. et al. (2016). Gated graph sequence neural networks. In *International Conference on Learning Representations (ICLR)*.

Li, Y. et al. (2018b). Factorizable Net: An Efficient Subgraph-based Framework for Scene Graph Generation. *European Conference on Computer Vision (ECCV)*.

Litjens, G. et al. (2017). A survey on deep learning in medical image analysis. In *Medical Image Analysis*, volume 42, pages 60–88.

Lu, M. et al. (2021a). Ai-based pathology predicts origins for cancers of unknown primary. In *Nature*.

Lu, M. et al. (2021b). Data efficient and weakly supervised computational pathology on whole slide images. In *Nat Biomed Eng*.

Lu, W. et al. (2020). Capturing cellular topology in multi-gigapixel pathology images. In *CVPR-W*.

Luo, D. et al. (2020). Parameterized explainer for graph neural network. In *Advances in neural information processing systems*.

Macenko, M. et al. (2009). A method for normalizing histology slides for quantitative analysis. In *IEEE International Symposium on Biomedical Imaging (ISBI)*, pages 1107–1110.

Madabhushi, A. and Lee, G. (2016). Image analysis and machine learning in digital pathology: Challenges and opportunities. In *Medical Image Analysis*, volume 33, pages 170–175.

Marami, B. et al. (2018). Ensemble network for region identification in breast histopathology slides. In *International Conference Image Analysis and Recognition*, pages 861–868.

Marcel, S. et al. (2010). Torchvision the machine-vision package of torch. In *Proceedings of the 18th ACM International Conference on Multimedia*, MM '10, page 1485–1488, New York, NY, USA. Association for Computing Machinery.

Mehta, S. et al. (2018). Learning to segment breast biopsy whole slide images. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*.

Mercan, C. et al. (2019a). From patch-level to roi-level deep feature representations for breast histopathology classification. In *SPIE Medical Imaging 2019: Digital Pathology*, volume 10956, page 109560H.

Mercan, E. et al. (2018). Automated diagnosis of breast cancer and pre-invasive lesions on digital whole slide images. In *ICPRAM*.

Mercan, E. et al. (2019b). Assessment of machine learning of breast pathology structures for automated differentiation of breast cancer and high-risk proliferative lesions. In *JAMA Netw Open*, volume 2.

Mohseni, S. et al. (2018). A multidisciplinary survey and framework for design and evaluation of explainable AI systems. In *arXiv:1811.11839*.

Montavon, G. et al. (2015). Explaining Non Linear Classification Decisions with Deep Taylor Decomposition. *Pattern Recognition*, 65:211–222.

Morris, C. et al. (2018). Weisfeiler and leman go neural: Higher-order graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33.

Moulin, P. et al. (2021). Imi—bigpicture: A central repository for digital pathology. In *Journal of Toxicologic Pathology*.

Mukhopadhyay, S. et al. (2017). Whole slide imaging versus microscopy for primary diagnosis in surgical pathology: a multicenter blinded randomized noninferiority study of 1992 cases (pivotal study). In *Am J Surg Pathol*, volume 42, pages 39–52.

Nguyen, A. and Rodriguez Martinez, M. (2020). On quantitative aspects of model interpretability. In *arXiv:2007.07584*.

Nguyen, L. et al. (2017). Achitectural patterns for differential diagnosis of proliferative breast lesions from histopathological images. In *IEEE International Symposium on Biomedical Imaging*, pages 152–155.

Niepert, M. et al. (2016). Learning convolutional neural networks for graphs. In *International Conference on Machine Learning (ICML)*, pages 2014–2023.

Orsini, F. et al. (2016). Graph invariant kernels. In *International Joint Conference on Artificial Intelligence (IJCAI)*.

Ortega, A. et al. (2018). Graph signal processing: Overview, challenges, and applications. *Proceedings of the IEEE*.

Ozen, Y. et al. (2020). Self-supervised learningwith graph neural networksfor region of interest retrieval in histopathology. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*.

Ozkan, T. A. et al. (2016). Interobserver variability in gleason histological grading of prostate cancer. *Scandinavian journal of urology*, 50(6):420–424.

Parwani, A. (2019). Next generation diagnostic pathology: use of digital pathology and artificial intelligence tools to augment a pathological diagnosis. In *Diagnostic Pathology*, volume 14.

Paszke, A. et al. (2019). Pytorch: An imperative style, high-performance deep learning library. In *Neural Information Processing Systems (NeurIPS)*, pages 8024–8035.

Pati, P. et al. (2018). Deep positive-unlabeled learning for region of interest localization in breast tissue images. In *SPIE Medical Imaging 2018: Digital Pathology*, volume 10581, page 1058107.

Pati, P. et al. (2020). Hact-net: A hierarchical cell-to-tissue graph neural network for histopathological image classification. In *Medical Image Computing and Computer Assisted Intervention (MICCAI) Workshop on GRaphs in biomedicAl Image anaLysis*.

Pati, P. et al. (2021a). Hierarchical graph representations in digital pathology. In *Medical Image Analysis*.

Pati, P. et al. (2021b). Reducing annotation effort in digital pathology: A co-representation learning framework for classification tasks. In *Medical Image Analysis*, volume 67.

Patterson, E. M. et al. (1967). *Modern Algebra (Prentice-Hall, Inc., 1965), two volumes, 806 pp.*, volume 15. Cambridge University Press.

Pinckaers, H. et al. (2020). Streaming convolutional neural networks for end-to-end learning with multi-megapixel images. In *IEEE Transactions on Medical Imaging*, volume 39, pages 1306–1315.

Pope, P. et al. (2019). Explainability methods for graph convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10764–10773.

Potjer, F. (1996). Region adjacency graphs and connected morphological operators. In *Mathematical Morphology and its Applications to Image and Signal Processing. Computational Imaging and Vision*, volume 5, page 111–118.

Rajbongshi, N. et al. (2018). Analysis of Morphological Features of Benign and Malignant Breast Cell Extracted From FNAC Microscopic Image Using the Pearsonian System of Curves. *Journal of Cytology*, 35(2):99–104.

Raju, A. et al. (2020). Graph attention multi-instance learning for accurate colorectal cancer staging. In *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*.

Rakha, E. et al. (2008). Prognostic significance of nottingham histologic grade in invasive breast carcinoma. In *Journal of Clinical Oncology*.

Reinhard, E. et al. (2001). Color transfer between images. In *Computer Graphics and Applications*.

Ren, J. et al. (2019). Unsupervised domain adaptation for classification of histopathology whole-slide images. *Frontiers in Bioengineering and Biotechnology*, 7:102.

Ren, S. et al. (2017). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149.

Ribeiro, M. et al. (2016). Why should i you? Explaining the predictions of any classifier. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1135–1144.

Ristoski, P. et al. (2016). Rdf2vec: Rdf graph embeddings for data mining. In *Lecture Notes in Computer Science*.

Roy, K. et al. (2019). Patch-based system for classification of breast histology images using deep learning. In *Computerized Medical Imaging and Graphics*, volume 71, pages 90–103.

Ruchika, G. et al. (2020). Multi-organ nuclei segmentation and classification challenge 2020. *IEEE Transactions on Medical Imaging*.

Salmo, E. N. (2015). An audit of inter-observer variability in gleason grading of prostate cancer biopsies: The experience of central pathology review in the north west of england. *Integr Cancer Sci Ther*, 2(2):104–106.

Samek, W. et al. (2017). Evaluating the visualization of what a deep neural network has learned. In *IEEE Transactions on Neural Networks and Learning Systems*, volume 28, pages 2660–2673.

Sandler, M. et al. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *IEEE CVPR*, pages 4510–4520.

Schlichtkrull, M. et al. (2018). Modeling relational data with graph convolutional networks. In *Extended Semantic Web Conference*.

Schlichtkrull, M. et al. (2021). Interpreting graph neural networks for nlp with differentiable edge masking. In *International Conference on Learning Representations (ICLR)*.

Schwaller, P. et al. (2018). "Found in Translation": predicting outcomes of complex organic chemistry reactions using neural sequence-to-sequence models† . *Chem. Sci.*

Schwarzenberg, R. et al. (2019). Layerwise relevance visualization in convolutional text graph classifiers. *EMNLP Workshop*, pages 58–62.

Selvaraju, R. et al. (2017). Grad-CAM : Visual Explanations from Deep Networks. In *International Conference on Computer Vision*, pages 618–626.

Serag, A. et al. (2019). Translational ai and deep learning in diagnostic pathology. In *Frontiers in Medicine*.

Shaban, M. et al. (2020). Context-aware convolutional neural network for grading of colorectal cancer histology images. In *IEEE Transactions on Medical Imaging*, volume 39, pages 2395 – 2405.

Sharma, H. et al. (2015). A review of graph-based methods for image analysis in digital histopathology. In *Diagnostic Pathology*.

Sharma, H. et al. (2016). Cell nuclei attributed relational graphs for efficient representation and classification of gastric cancer in digital histopathology. In *SPIE Medical Imaging: Digital Pathology*, volume 9791.

Sharma, H. et al. (2017). A comparative study of cell nuclei attributed relational graphs for knowledge description and categorization in histopathological gastric cancer whole slide images. In *IEEE Symposium on Computer-Based Medical Systems*, pages 61–66.

Sharma, M. et al. (2020). Percent gleason pattern 4 in stratifying the prognosis of patients with intermediate-risk prostate cancer. *Transl Androl Urol*.

Shervashidze, N. et al. (2011). Weisfeiler-Lehman Graph Kernels. *Journal of Machine Learning Research*, 12:2539–2561.

Shi, Y. et al. (2020). Building segmentation through a gated graph convolutional neural network with deep structured feature embedding. In *ISPRS Journal of Photogrammetry and Remote Sensing*, volume 159, pages 184–197.

Shuman, D. I. et al. (2013). The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE Signal Processing Magazine*.

Siegel, R. et al. (2020). Cancer statistics, 2020. In *CA: A Cancer Journal for Clinicians*, volume 70, pages 7–30.

Silva-Rodríguez, J. et al. (2021). Weglenet: A weakly-supervised convolutional neural network for the semantic segmentation of gleason grades in prostate histology images. *Computerized Medical Imaging and Graphics*, 88:101846.

Silva-Rodríguez, J. et al. (2020). Going deeper through the gleason scoring scale: An automatic end-to-end system for histology prostate grading and cribriform pattern detection. In *Computer Methods and Programs in Biomedicine*, volume 195.

Simonovsky, M. et al. (2017). Dynamic edge-conditioned filters in convolutional neural networks on graphs. In *Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Simonyan, K. et al. (2013). Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. *arXiv:1312.6034*.

Sirinukunwattana, K. et al. (2017). Gland segmentation in colon histology images: The glas challenge contest. In *Medical Image Analysis*, volume 35, pages 489–502.

Sirinukunwattana, K. o. (2018). Improving whole slide segmentation through visual context - a systematic study. In *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, volume 11071.

Spanhol, F. et al. (2016). A dataset for breast cancer histopathological image classification. In *IEEE Transactions on Biomedical Engineering*, volume 63, pages 1455–1462.

Srinidhi, C. et al. (2021). Deep neural network models for computational histopathology: A survey. In *Medical Image Analysis*, volume 67.

Stanisavljevic, M. et al. (2018). A fast and scalable pipeline for stain normalization of whole-slide images in histopathology. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*.

Stokes, J. M. et al. (2020). A deep learning approach to antibiotic discovery. *Cell*, 180(4):688–702.e13.

Ström, P. et al. (2019). Pathologist-level grading of prostate biopsies with artificial intelligence. In *Bioinformatics*.

Studer, L. et al. (2021). Classification of intestinal gland cell-graphs using graph neural networks. In *International Conference on Pattern Recognition (ICPR)*.

Sureka, M. et al. (2020). Visualization for histopathology images using graph convolutional neural networks. In *arXiv:2006.09464*.

Tan, P. et al. (2019). The 2019 world health organization classification of tumours of the breast. In *International Agency for Research on Cancer*, volume 5.

Tellez, D. et al. (2018). H and e stain augmentation improves generalization of convolutional networks for histopathological mitosis detection. In *Medical Imaging*.

Tellez, D. et al. (2019a). Neural image compression for gigapixel histopathology image analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 58.

Tellez, D. et al. (2019b). Quantifying the effects of data augmentation and stain color normalization in convolutional neural networks for computational pathology. In *Medical Image Analysis*, volume 58.

Thagaard, J. et al. (2020). Can you trust predictive uncertainty under real dataset shifts in digital pathology? In *Medical Image Computing and Computer Assisted Intervention (MICCAI)*.

Thai-Nghe, N. et al. (2010). Cost-sensitive learning methods for imbalanced data. In *The 2010 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8.

Tizhoosh, H. and Pantanowitz, L. (2018). Artificial Intelligence and Digital Pathology: Challenges and Opportunities. *Journal of Pathology Informatics*, 38(9).

Vahadane, A. et al. (2016). Structure-preserving color normalization and sparse stain separation for histological images. In *IEEE Transactions on Medical Imaging*, volume 35, pages 1962–1971.

Van der Laak, J. et al. (2021). Deep learning in histopathology: the path to the clinic. In *Nature Medicine*.

Van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Velickovic, P. et al. (2018). Graph attention networks. In *International Conference on Learning Representations (ICLR)*.

Verma, R. et al. (2021). Multi-organ nuclei segmentation and classification challenge 2020. *IEEE Transactions on Medical Imaging*, 39:1380–1391.

Veta, M. et al. (2014). Breast cancer histopathology image analysis: A review. In *IEEE Transactions on Biomedical Engineering*.

Vu, M. et al. (2020). Pgm-explainer: Probabilistic graphical model explanations for graph neural networks. In *Advances in neural information processing systems*.

Wang, J. et al. (2020). Weakly su-pervised prostate tma classification via graph convolutional networks. In *Proceedings of IEEE International Symposium on Biomedical Imaging (ISBI)*.

Wang, M. et al. (2019a). Deep graph library: Towards efficient and scalable deep learning on graphs. In *CoRR*, volume abs/1909.01315.

Wang, S. et al. (2019b). Pathology image analysis using segmentation deep learning algorithms. In *The American Journal of Pathology*, volume 189, pages 1686–1698.

Weininger, D. et al. (1988). SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *J. Chem. Inf. Comput. Sci.*

Weisfeiler, B. and Lehman, A. A. (1968). A reduction of a graph to a canonical form and an algebra arising during this reduction. In *Nauchno-Technicheskaya Informatsia*, volume 2, pages 12–16.

Wilson, M. W. et al. (2018). Access to pathology and laboratory medicine services: A crucial gap. In *Lancet*.

Wu, J. et al. (2019). Weakly-and semi-supervised graph cnn for identifying basal cell carcinomaon pathological images. In *Proceedings of the International Workshop Graph Learning in Medical Imaging (GLMI)*.

Wu, Z. et al. (2020). A Comprehensive Survey on Graph NeuralNetworks. *IEEE Transactions on Neural Networks and Learning Systems*.

Xie, J. et al. (2019). Deep learning based analysis of histopathological images of breast cancer. In *Frontiers in Genetics*.

Xu, D. et al. (2017a). Scene graph generation by iterative message passing. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Xu, G. et al. (2019a). Camel: A weakly supervised learning framework for histopathology image segmentation. In *IEEE ICCV*, pages 10681–10690.

Xu, K. et al. (2018). Representation learning on graphs with jumping knowledge networks. In *International Conference on Machine Learning (ICML)*.

Xu, K. et al. (2019b). How powerful are graph neural networks? In *International Conference on Learning Representations (ICLR)*.

Xu, Y. et al. (2014). Weakly supervised histopathology cancer image segmentation and classification. In *Medical Image Analysis*, volume 18, pages 591–604.

Xu, Y. et al. (2017b). Large scale tissue histopathology image classification, segmentation, and visualization via deep convolutional activation features. In *BMC bioinformatics*, volume 18.

Yan, R. et al. (2020). Breast cancer histopathological image classification using a hybrid deep neural network. In *Methods*, volume 173, pages 52–60.

Ying, R. et al. (2019). GNNExplainer: Generating Explanations for Graph Neural Networks. In *Advances in Neural Information Processing Systems*.

Yosinski, J. et al. (2015). Understanding neural networks through deep visualization. *International Conference on Machine Learning Workshops*.

Yuan, H. et al. (2020). On explainability of graph neural networks via subgraph explorations. In *arXiv preprint arXiv:2102.05152*.

Yuan, H. et al. (2021). Explainability in graph neural networks: A taxonomic survey. In *arXiv preprint arXiv:2012.15445*.

Zeiler, M. and Fergus, R. (2014). Visualizing and understanding convolutional networks. *European Conference on Computer Vision*.

Zellers, R. et al. (2017). Neural Motifs: Scene Graph Parsing with Global Context. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhang, M. et al. (2020). Feature-enhanced graph networks for genetic mutational prediction using histopathological images in colon cancer. In *ArXiv preprint arXiv:2012.14619*.

Zhang, Z. et al. (2017). MDNet: A Semantically and Visually Interpretable Medical Image Diagnosis Network. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Zhang, Z. et al. (2019). Pathologist-level interpretable whole-slide cancer diagnosis with deep learning. *Nature Machine Intelligence*, pages 236–245.

Zhao, Y. et al. (2020). Predicting lymph node metastasis using histopathological images based on multiple instance learning with deep graph convolution. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4837–4846.

Zheng, Y. et al. (2019). Encoding histopathological wsis using gnn for scalable diagnostically relevant regions retrieval. In *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*.

Zhong, Q. et al. (2017). A curated collection of tissue microarray images and clinical outcome data of prostate cancer patients. In *Scientific Data*, volume 4.

Zhou, B. et al. (2016). Learning deep features for discriminative localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Zhou, J. et al. (2021). Graph Neural Networks: A Review of Methods and Applications. *AI Open*.

Zhou, Y. et al. (2019a). CGC-net: Cell graph convolutional network for grading of colorectal cancer histology images. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops*.

Zhou, Z. et al. (2019b). Deep Learning on Graphs: A Survey. *arXiv 1812.04202*.

Zintgraf, L. M., Cohen, T. S., Adel, T., and Welling, M. (2017). Visualizing deep neural network decisions: Prediction difference analysis. *International Conference on Learning Representations*.

Zitnik, M. et al. (2018). Modeling polypharmacy side effects with graph convolutional networks. In *Bioinformatics*.

Guillaume Jaume

☎ +41-78-661-4152 ✉ guillaume.jaume@epfl.ch guillaume-jaume Guillaume Jaume guillaumejaume

RESEARCH INTERESTS

My research focuses on deep learning for graph-structured data with applications to computational pathology. I developed graph-based representations and models of histopathological tissues, notably by leveraging Graph Neural Networks. Specifically, I have explored three lines of research in computational pathology: scalability, explainability and weakly supervised settings.

EDUCATION

- **Ph.D. in Electrical Engineering** *Jan 2018 - Jan 2022*
  IBM Research, Zurich & EPFL, Lausanne, Switzerland
  Advisors: Prof. Dr. Jean-Philippe Thiran; Dr. Maria Gabrani
- **M.Sc. in Electrical Engineering & Information Technology** *Sep 2015 - Sep 2017*
  EPFL, Lausanne, Switzerland
  Thesis: A Cognitive Solution to Extract and Understand Information in Medical Forms (GPA 6/6)
- **Erasmus exchange, Electrical & Computer Engineering** *Sep 2014 - June 2015*
  Heriot-Watt University, Edinburgh, United Kingdom
- **B.Sc. in Electrical Engineering** *Sep 2012 - June 2015*
  EPFL, Lausanne, Switzerland

WORK EXPERIENCE

- **IBM Research Zurich, Switzerland** *Dec 2017 - Present*
  Pre-doctoral researcher in the Cognitive Healthcare & Life Sciences group
  Focus: Computational Pathology, Graph Representation Learning
  Collaborators: ETH Zurich, CHUV Lausanne, University Hospital of Zurich, University Hospital of Paris, University of Bern, Istanbul Technical University, National Research Council of Italy
- **EPFL, Lausanne, Switzerland** *Sep 2014 - Jun 2016*
  Teaching Assistant with Prof. Dr. Nicolas Macris & Prof. Dr. Andreas Burg
  Focus: Supervise students in practicals, projects and labs
- **CERN, Geneva, Switzerland** *June 2015 - Aug 2015*
  CERN Summer Student Program, High-Luminosity LHC
  Project: Development of 3D automation tools for Radio Frequency measurements

PUBLICATIONS

Journals:

- P. Pati*, **G. Jaume***, A. Foncubierta-Rodriguez et al., "Hierarchical Graph Representations in Digital Pathology," Medical Image Analysis, 2021 [arXiv] [Code]

Conferences & Workshops:

- **G. Jaume***, P. Pati*, B. Bozorgtabar et al., "Quantifying Explainers of Graph Neural Networks in Computational Pathology," IEEE CVPR, 2021 [arXiv] [Code]
- V. Anklin*, P. Pati*, **G. Jaume*** et al., "Learning Whole-Slide Segmentation from Inexact and Incomplete Labels using Tissue Graphs," MICCAI, 2021 [arXiv] [Code]

*denotes equal contribution

- **G. Jaume***, P. Pati*, A. Foncubierta-Rodriguez et al., "HistoCartography: A Toolkit for Graph Analytics in Digital Pathology," MICCAI Compay Workshop, 2021 [arXiv] [Code] **[Best Software Paper Award]**
- P. Pati*, **G. Jaume***, A. Foncubierta-Rodriguez et al., "HACT-Net: A Hierarchical Cell-to-Tissue Graph Neural Network for Histopathological Image Classification," MICCAI, Graphs in Biomedical Image Analysis Workshop, 2020 [arXiv] **[Best paper award]**
- **G. Jaume***, P. Pati*, A. Foncubierta-Rodriguez et al., "Towards Explainable Graph Representations in Digital Pathology," ICML, Computational Biology Workshop, 2020 [arXiv] **[Best paper award]**
- **G. Jaume**, H. Ekenel, J-P. Thiran, "FUNSD: A Dataset for Form Understanding in Noisy Scanned Documents," IEEE ICDAR, 2019 [arXiv] [Website]
- **G. Jaume**, A. Nguyen, M. Martinez et al., "edGNN: A simple and powerful GNN for labeled graphs," ICLR, Representation Learning on Graphs and Manifolds Workshop, 2019 [arXiv] [Code]
- **G. Jaume**, B. Bozorgtabar, H. Ekenel et al., "Image-Level Attentional Context Modeling Using Nested-Graph Neural Networks," NeurIPS, Relational Representation Learning Workshop, 2018 [arXiv]

Book chapters:

- P. Pati*, **G. Jaume***, A. Foncubierta-Rodriguez, et al., "Graph Representation Learning & Explainability in Breast Cancer Pathology: Bridging the gap between AI and Pathology Practice," Artificial Intelligence as applied in Human Pathology, Editor: R. Huss, World Scientific, 2021

Preprints:

- **G. Jaume***, P. Pati*, et al., "Weakly Supervised Learning for Joint Whole-Slide Segmentation and Classification in Prostate Cancer," 2021
- N. Brancati,..., **G. Jaume**, et al., "BRACS: A Dataset for BReAst Carcinoma Subtyping in H&E Histology Images," 2021

LIBRARY & DATASETS

- **HistoCartography**: A collection of image-to-graph translation and state-of-the-art graph algorithms for facilitating interpretable entity-based analysis in digital pathology [Code]
- **BR**e**A**st **C**arcinoma **S**ubtyping (**BRACS**): A large cohort of H&E stained histopathological images for automated breast cancer diagnosis [Website]
- **FUNSD**: A dataset for Form Understanding in Noisy Scanned Documents [Website]

PATENTS

- P. Pati, **G. Jaume**, K. Thandiackal, A. Foncubierta-Rodriguez, M. Gabrani, "Registration Free Multimodal Digital Pathology," 2021 [Filed]
- P. Pati, **G. Jaume**, A. Foncubierta-Rodriguez, M. Gabrani, "AI agent to assist pathology whole slide image interpretation through hierarchical representations," 2021 [Filed]
- **G. Jaume**, A. Foncubierta-Rodriguez, M. Gabrani, "Extracting structured information from a document containing filled form images," 2019 [Granted]
- **G. Jaume**, A. Foncubierta-Rodriguez, M. Gabrani, "Method and system for extracting information from an image of a filled form document," 2019 [Granted]

AWARDS

- IBM Outstanding Technical Achievement and Innovation Award *May 2021*
  "Intelligent and quantitative immunostaining of tumor tissue sections"

- IBM First Invention Plateau *June 2021*
- Best Paper Awards:
  - MICCAI, Computational Pathology (COMPAY) Workshop *Sep 2021*
  - MICCAI, Graphs in Biomedical Image Analysis Workshop *Oct 2020*
  - ICML, Computational Biology Workshop *July 2020*

STUDENT SUPERVISION

- Valentin Anklin, *Master's thesis* *Autumn 2020*
  "Learning Segmentation in Histology from Inexact and Incomplete Labels using GNNs"
- Lauren Alisha Fernandez, *Master's thesis* *Autumn 2019*
  "Cell-graph Networks for Representation and Grading of Histopathology Images"
- Atul Kumar, *Master's thesis* *Autumn 2019*
  "Learning to generate Scene Graphs from Images and vice-versa"
- Martin Svatos, *Research internship* *Spring 2019*
  "Mind the Logit Gap: Incomparable Tasks in Continual Learning"
- Maria Halushko, *Research internship* *Autumn 2018*
  "Text Detection in Noisy Scanned Documents"

COMMUNITY SERVICE

- **Workshop Co-organizer:**
  - IEEE International Symposium on Biomedical Imaging (ISBI), *Kolkata* *March 2022*
    "BRIGHT: BReast tumor Image classification on Gigapixel HisTopathological images"
  - American Medical Informatics Association (AMIA), *San Diego* *Nov 2021*
    "Workshop on Explainable Multimodal AI in Cancer Patient Care"
  - Applied Machine Learning Days (AMLD), *Lausanne* *April 2021*
    "Building Interpretable AI for Digital Pathology" [Code]
- **Talks:**
  - Tissue Image Analytics Centre, *Warwick* – Invited by Prof. Nasir Rajpoot *Oct 2021*
    "HistoCartography: Graph representations and models in Computational Pathology"
  - Charité University Hospital, *Berlin* *Oct 2021*
    "Graph Representations and Models in Digital Pathology"
  - PathAI, *New York* *July 2021*
    "Weakly-Supervised Learning for Whole-Slide-Image Segmentation"
  - Harvard Medical School, *Boston* – Invited by Prof. Faisal Mahmood *July 2021*
    "A Graph Network Tour of Computational Pathology"
  - Lausanne University Hospital (CHUV), *Lausanne* *May 2021*
    "Computational Pathology: Building Interpretable AI at Scale"
  - Swiss Digital Pathology Consortium (SDiPath), *Bern* *Jan 2021*
    "Graph Representation Learning & Explainability in Computational Pathology"
  - Computer Research Institute of Montreal (CRIM), *Montreal* *Nov 2020*
    "Deep Learning on Graphs: An Overview" [Code]
  - 10+ Internal IBM Talks, *Zurich* *2019-2021*
    IBM Research, IBM Watson, IBM Global Business Services
- **Reviewer:** IEEE CVPR, Medical Image Analysis

REFERENCES

- Prof. Dr. Jean-Philippe Thiran jean-philippe.thiran@epfl.ch
  Full Professor, EPFL, Lausanne
- Prof. Dr. Inti Zlobek inti.zlobec@pathology.unibe.ch
  Head of Translational Research Unit, University of Bern
- Dr. Pierre Moulin pierre.moulin@novartis.com
  Project Lead IMI BigPicture, Novartis, Basel