EPFL

Thèse n° 10 296

Self-Supervised Learning for Patient Stratification and Survival Analysis in Computational Pathology: An Application to Colorectal Cancer

Présentée le 28 novembre 2023

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

Christian Robert ABBET

Acceptée sur proposition du jury

Prof. S. Carrara, président du jury
Prof. J.-Ph. Thiran, Prof. I. Zlobec, directeurs de thèse
Dr P. Moulin, rapporteur
Dr S. Kjær-Frifeldt , rapporteuse
Prof. P. Frossard, rapporteur

École polytechnique fédérale de Lausanne

2023

Jour meilleur

Allergique à la vie, les matins sont obscurs.
Quand tout a un arrière-goût de déjà vu.
Les nuits sont mortes, tout le monde t'a abandonné, même la lune.
Mais la fin du désert se cache peut-être derrière chaque dune.

Tout va s'arranger, c'est faux, je sais que tu le sais.
Des fois je ne saurai plus trop quoi dire, mais je pourrai toujours écouter.
Tout ne va pas changer, enfin, sauf si tu le fais.
Quand tu as le désert à traverser, il n'y a rien à faire, sauf d'avancer.

On en rira quand on le verra sous un jour meilleur.

Civilisation
Aurélien Cotentin

To my family, partner, and friends...

Acknowledgements

It is difficult for me to adequately express my gratitude toward those who helped and supported me. This journey started many years ago and finally reached its end. As with every path, it had its ups and downs. I sometimes regard my studies as an emotional roller coaster. Moments where I doubted my choices, others where I was proud of how far I had come.

First, I thank Prof. Jean-Philippe Thiran for allowing me to start this thesis. I first met him as a teacher in my first years at EPFL. Little did I expect I would end up rescuing bees with his help as a Master project. Over the years, he always trusted my work and left me independent in my research; I am deeply thankful for that. Second, I thank Prof. Inti Zlobec for her help throughout my thesis. Before starting my project, I knew little, if not nothing, about cancer and histology. She spent hours explaining to me in detail everything she knew about her field of research with great interest, kindness, and joy. For all of this, I am grateful to have you both as supervisors for my thesis.

Most of my days have been spent at EPFL in the LTS5. There, I had the opportunity to meet amazing people who went from colleagues to friends. Over the years, they had to endure my questionable jokes and complaints. I thank Alex, Anne, Benoit, Christophe, Davide, Devavrat, Guillaume, Juan-Luis, Magali, Pauline, Rémy, Roser, Samuel, Sandra, and Thomas. In addition to my time spent in Lausanne, I had the chance to travel to Bern to learn more about the clinical importance of my work. There, I would like to thank the equally amazing people as Amjad, Ana, Elias, Linda, Mauro, and Philipp, who accepted me as a member of their group even though I came from Romandy. Thanks to Linda, who spent countless hours and days proofreading our works and publications. Without your help, my publications list would probably have been empty. Thanks also to Davide, Magali, Philipp, and Rémy for proofreading this document.

Throughout my thesis, I also had the chance to rely on my longtime friends. Sometimes friendships start in simple ways, such as shared interests in *becoming the very best*, board games, or pub quizzes. They always supported me no matter the difficulty of the situation, and just for that, I owe you a lot. So to Didier, Emilie, Gaétan, Maxime and Nicolas thank you.

Lastly, I thank my family and partner for their unconditional support. They were there during my brightest but, most importantly, darkest times. When things seemed

unfeasible, they helped me withstand and go forward. To Murielle, Stéphanie, Papa, Maman, Grand-Maman, Nonno, Nonna, and Valentine, please accept my gratitude.

To all the people I mentioned and those I might have forgotten, this thesis is also yours. For your kindness, help, trust, confidence, and dedication, *merci du fond du coeur*.

Lausanne, November 3, 2023 Christian Abbet

Abstract

Over the years, clinical institutes accumulated large amounts of digital slides from resected tissue specimens. These digital images, called whole slide images (WSIs), are high-resolution tissue snapshots that depict the complex interaction of cells at the microscopic level. WSIs are critical to pathologists as they are used to identify disease status and target appropriate patient treatments. However, the abundance of WSIs comes with one primary drawback: the absence or scarcity of annotations. The accessibility to labeled data is usually limited to critical information such as the patient's clinical reports. The reason is that generating additional annotations is tedious and time-expensive for pathologists and, hence, should be avoided. Unfortunately, traditional supervised machine learning relies on fully labeled data to be trained, which is unavailable in this context. As a result, a significant part of the data ends up being discarded.

Out of the various approaches developed to tackle the inherent problem of label scarcity, self-supervised learning (SSL) appears as a viable solution. SSL is based on the supervision of data itself. In other words, it uses data structure as a pretext task to learn feature representations. Consequently, self-supervised approaches can take advantage of the broadly available clinical cohorts to train robust tissue descriptors without prior knowledge of data labels. SSL models are mainly used to initialize downstream tasks such as classification, segmentation, or survival analysis. Downstream tasks initialized with pre-trained models generally require few labeled data to converge to optimal solutions, thus reducing the impact of label sparsity.

Unfortunately, learning tissue representation from pathological data itself is challenging. WSIs include various structural and visual biases that can hinder the performance of our pre-trained models. For example, data acquired from different institutes might show visual differences in staining intensity. This discrepancy appears as a strong domain shift in the learned feature space, which makes pre-trained models less efficient for inter-clinical applications. Another critical aspect is the inherent data complexity and heterogeneity, which is not reflected in publicly available cohorts. These are often composed of curated data that represent homogeneous tissue structures. This asymmetry can harm the quality of tissue segmentation in downstream tasks and clinical metrics assessment.

In this thesis, we address the mentioned issues on computation pathology and label availability. We propose novel approaches that take advantage of SSL to learn and build

complex tissue descriptors while avoiding access to labeled data. More specifically, we first present a simple way to benefit from WSIs staining information to learn robust feature spaces using SSL. Secondly, we tackle the problem of domain shift and data heterogeneity by allowing the use of multi-source data to strengthen the quality of feature representation. Next, we investigate the limitations of SSL when applied to tissue segmentation and propose an alternative based on coarsely annotated data. Finally, we conclude this work by building clinically relevant metrics based on our previously designed architectures. By doing so, we aim to demonstrate the applicability of our research by creating a bridge between theory and practice.

Keywords: *Computer Vision, Machine Learning, Digital Pathology, Computational Pathology, Self-supervised Learning, Label Scarcity, Survival Analysis.*

Résumé

Au cours des années, les instituts cliniques ont accumulé de grandes quantités d'images numériques provenant d'échantillons de tissus réséqués. Ces images numériques, appelées whole slide images (WSIs) en anglais, sont des "photographies" de tissus à haute résolution qui décrivent l'interaction complexe des cellules au niveau microscopique. Ces diapositives sont essentielles pour les pathologues, car elles permettent d'identifier le statut de la maladie et de cibler les traitements appropriés. Cependant, l'abondance de WSIs s'accompagne d'un inconvénient majeur, à savoir l'absence ou la rareté des annotations. L'accès aux données annotées est généralement limité aux informations critiques telles que les rapports cliniques des patients. La raison en est que la génération d'annotations supplémentaires est fastidieuse et coûteuse en temps pour les pathologues et doit donc être évitée autant que possible. Malheureusement, le traditionnel apprentissage automatique supervisé repose sur des données entièrement annotées, qui ne sont pas disponibles dans ce contexte. Par conséquent, une partie importante des données finit par être écartée.

Parmi les différentes approches développées pour résoudre le problème inhérent à la rareté des annotations, l'apprentissage auto-supervisé (ou self-supervised learning (SSL) en anglais) apparaît comme une solution viable. Le SSL est basé sur la supervision des données elles-mêmes. En d'autres termes, il utilise la structure des données comme tâche préalable à l'apprentissage de ses propres caractéristiques. Par conséquent, les approches auto-supervisées peuvent tirer parti des cohortes cliniques disponibles pour former des descripteurs tissulaires robustes sans connaissance préalable des annotations. Les modèles SSL sont principalement utilisés comme initialisation pour des tâches concrètes telles que la classification, la segmentation ou l'analyse de survie. Ces tâches qui sont initialisées avec des modèles pré-entraînés nécessitent généralement peu de données annotées pour être entraînées, ce qui réduit l'impact de la rareté de celles-ci.

Malheureusement, apprendre la représentation des tissus à partir des données pathologiques n'est pas une tâche triviale. Les WSIs comprennent divers biais structurels et visuels qui peuvent entraver la performance des modèles pré-entraînés. Par exemple, les données acquises dans différents instituts peuvent présenter des différences visuelles en termes d'intensité de coloration. Ces divergences se manifestent par un décalage dans la distribution des valeurs apprises, ce qui rend les modèles pré-entraînés moins efficaces pour les applications intercliniques. Un autre aspect critique est l'hétérogénéité

des données cliniques. Les cohortes accessibles publiquement sont souvent composées de données épurées qui représentent des structures tissulaires homogènes. Cependant, dans la pratique, les tissus sont des milieux complexes et hétérogènes. Cette asymétrie peut nuire à la qualité de la segmentation des tissus. Elle est d'autant plus problématique que la qualité de la segmentation est essentielle pour l'évaluation de certains paramètres cliniques et par conséquent pour l'analyse de survie.

Dans cette thèse, nous abordons les limitations liées à la disponibilité des annotations. Nous proposons de nouvelles approches qui tirent parti du SSL pour apprendre et construire des descripteurs tissulaires complexes tout en évitant l'accès aux données annotées. Plus précisément, nous présentons d'abord un moyen simple de tirer parti des informations de coloration des WSIs pour apprendre des caractéristiques cohérentes. Ensuite, nous nous attaquons au problème du décalage de domaine pour permettre l'utilisation de données multi-sources. Par le suite, nous étudions les limites du SSL lorsqu'il est appliqué à la segmentation des tissus et proposons une alternative basée sur des données grossièrement annotées. Enfin, nous concluons ce travail en construisant des métriques cliniquement pertinentes basées sur les architectures précédemment conçues. Ce faisant, nous visons à démontrer l'applicabilité clinique de notre recherche en créant un pont entre la théorie et la pratique.

Mots clés : *Vision par ordinateur, apprentissage automatique, pathologie numérique, pathologie informatique, apprentissage auto-supervisé, rareté des annotations, analyse de survie.*

Sommario

Nel corso delgli anni, gli istituti clinici hanno accumulato grandi quantità di vetrini digitali provenienti da campioni di tessuto asportati. Queste immagini digitali, chiamate whole slide images (WSIs) in inglese, sono “fotografie” di tessuto ad alta risoluzione che raffigurano la complessa interazione delle cellule a livello microscopico. Questi vetrini sono fondamentali per i patologi, in quanto vengono utilizzati per identificare lo stato della malattia e individuare i trattamenti appropriati per i pazienti. Tuttavia, l'abbondanza di dati ha uno svantaggio principale : l'assenza o la scarsità di annotazioni. L'accessibilità ai dati annotati è di solito limitata a informazioni critiche come il referto clinico del paziente. Ciò è dovuto al fatto che la generazione di annotazioni aggiuntive è tediosa e costosa per i patologi e quindi dovrebbe essere evitata. Purtroppo, il tradizionale apprendimento automatico supervisionato si basa su dati completamente annotati per l'addestramento, che non sono disponibili in questo contesto. Di conseguenza, una parte significativa dei dati finisce per essere scartata.

Tra i vari approcci sviluppati per affrontare il problema intrinseco della scarsità di annotazione, l'apprendimento auto-supervisionato, o self-supervised learning (SSL) in inglese, appare come una soluzione praticabile. Il metodo SSL si basa sulla supervisione da parte dei dati stessi. In altre parole, utilizza la struttura dei dati come compito preliminare per apprendere la rappresentazione delle loro caratteristiche. Di conseguenza, gli approcci auto-supervisionati possono sfruttare le coorti cliniche ampiamente disponibili per addestrare robusti descrittori tissutali senza una conoscenza preliminare delle annotazione dei dati. I modelli SSL sono usati principalmente come inizializzazione per compiti a valle, come la classificazione, la segmentazione o l'analisi di sopravvivenza. I compiti a valle che vengono inizializzati con modelli pre-addestrati richiedono in genere pochi dati annotati per essere addestrati, riducendo così l'impatto della loro scarsità.

Purtroppo l'apprendimento della rappresentazione dei tessuti dai dati patologici non è un compito banale. I WSIs comprendono diverse variazioni strutturale e visive che possono ostacolare le prestazioni dei nostri modelli preaddestrati. Ad esempio, i dati acquisiti da istituti diversi potrebbero mostrare differenze visive in termini di intensità di colorazione. Questa discrepanza si manifesta come un forte scostamento del dominio nello spazio delle caratteristiche apprese, che rende i modelli pre-addestrati meno efficaci per le applicazioni inter-cliniche. Un altro aspetto critico è l'eterogeneità dei dati. Le coorti

disponibili pubblicamente sono spesso composte da dati selezionati che rappresentano strutture tissutali omogenee. Tuttavia, nella pratica, i tessuti sono struture complesse ed eterogene. Questa asimmetria può compromettere la qualità della segmentazione dei tessuti nelle applicazioni reali. È ancora più problematica perché le segmentazione a grana fine sono essenziali per la valutazione delle metriche cliniche e quindi per l'analisi della sopravvivenza.

In questa tesi, cerchiamo di affrontare le problematiche menzionate sulla patologia computazionale e sulla disponibilità di annotazione. Proponiamo approcci innovativi che sfruttano le informazioni di l'SSL per apprendere e costruire descrittori tissutali complessi evitando l'accesso a dati annotati. Più specificamente, presentiamo innanzitutto un modo semplice per trarre vantaggio dalle informazioni di colorazione del WSIs per apprendere spazi di caratteristiche robusti. In secondo luogo, affrontiamo il problema del scostamento di dominio e della eterogeneità dei data consentendo l'uso di dati provenienti da più fonti, rafforzando così la qualità della rappresentazione delle caratteristiche. Successivamente, analizziamo le limitazioni del SSL quando viene applicato alla segmentazione dei tessuti e proponiamo un'alternativa basata su dati annotati in modo grossolano. Infine, concludiamo questo lavoro costruendo una metrica clinicamente rilevante basata sulle architetture precedentemente progettate. In questo modo, intendiamo dimostrare l'applicabilità della nostra ricerca creando un ponte tra teoria e pratica.

Parole chiave : *Computer Vision, Machine Learning, Patologia digitale, Patologia computazionale, Apprendimento auto-supervisionato, Scarsità di etichette, Analisi della sopravvivenza.*

Contents

List of Figures

List of Tables

Symbols and Notations

Mathematical

Matrices

$\mathbf{X}$	Input RGB image $\mathbf{X} \in \mathbb{R}^{W \times H \times 3}$, where H, W are the heigth and width of the image, respectively.
$(x_{i,j})_{1 \leq i,j \leq N}$	Matrix of dimensions $N \times N$ and entries $x_{i,j}$.
$(\mathbf{X})_{i,j}$	Spatial entry (i, j) of the image $\mathbf{X}$.
$\mathbf{X}_i$	i-th image X.
$\mathbf{X}^{HE}$	Hematoxylin & Eosin channels of an image $\mathbf{X}$.
$\mathbf{X}^{E}$	Eosin channel of an image $\mathbf{X}$.
$\mathbf{X}^{H}$	Hematoxylin channel of an image $\mathbf{X}$.

Vectors

$\mathbf{z}$	Feature representation.
$\mathbf{z}_i$	Feature representation of i-th tile $\mathbf{X}_i$.
$\widetilde{\mathbf{z}}$	Memory bank entry.

Operators & Functions

ζ	Estimation of hematoxylin and eosin stain.
τ	Image color normalization.
ξ	Apply a set of transformations to input tiles.
$\cap$	Intersection of sets or masks.
$*$	Convolution operator.
$\cup$	Union of sets or masks.
$\langle \cdot\ , \cdot \rangle$	Dot product between vectors.
$\setminus$	Exclude part of the set.
f_θ	Learnable model f with parameters θ.
arg	Argument (angle) of a complex number.
arctan2	Measure of a vector angle with respect to the origin as $\theta = \arctan(x/y)$.

coord	2D coordinates (x, y) of a tile within a whole slide image.
cov	Covariance of a matrix.
dist	Euclidean distance between two 2D points.
vec	Flatten a matrix into a vector.
$\lfloor \cdot \rfloor$	Gives the largest integer less than or equal to the variable.
$\mathbb{1}_A$	Indicator function. It is equal to 1 if the condition A is matched, 0 otherwise.
$\{\cdot\}_A$	Masking function. Returns the indexed entries of a matrix matching condition A.
Q_α	Compute the α-th quantile of a vector.
Π	Rectangular function that is equal to 1 if evaluated in the range $[-\frac{1}{2}, \frac{1}{2}]$ and 0 elsewhere.

Sets

$\mathcal{D}^s$	Source set of tiles.
$\mathcal{D}$	Set of data.
$\{\mathbf{x}_i\}_{i=1}^N$	Set of vectors $\mathbf{x}_i$ where $i \in \{1, \ldots, N\}$.
$\mathcal{D}_k^s$	k-th source tiles dataset.
$\mathcal{D}^t$	Target tiles dataset.
$\mathcal{N}$	Neighborhood in the feature domain.
$\mathcal{P}_\mathcal{A}$	Patient set $\mathcal{A}$ with clinical data.
$\mathcal{Q}$	Queue embedding set.
$\lvert\mathcal{S}\rvert$	Number of elements in set $\mathcal{S}$.
$\mathcal{S}$	Neighborhood in the spatial domain.
$\mathcal{W}$	Whole slide image composed of $\mathbf{X}$ tiles.
$\mathcal{Z}$	Set of feature embedding or memory bank.

Miscellaneous

$\mathcal{L}$	Loss term to optimize.
$a \propto b$	Indicates proportionality between a and b ($\frac{a}{b} = const$).

Acronyms

e.g.	*exempli gratia*
i.e.	*id est*
ADI	adipose
ADV	advent
BACK	background
BLOOD	blood
CSTR	complex stroma
DEB	debris
LYM	lymphocytes

MUC	mucin
MUS	muscle
NORM	normal mucosa
STR	stroma
TA-STR	tumor-associated stroma
TUM	tumor

BS	Brier score
BT	Barlow Twins
BYOL	*Bootstrap Your Own Latent*
C2R	coarse to refined
CAM	class activation map
CD	Chamfer distance
C-Index	concordance index
CI	confidence interval
CL	constrastive learning
CMS	consensus molecular subtypes
CNN	convolutional neural network
COAD	colon adenocarcinoma
CPC	contrastive predictive coding
CPH	Cox proportional hazards
CRCTP	colorectal cancer tissue phenotype
CRC	colorectal cancer
DAB	diaminobenzidine
DCA	clustering assignment
DCD	density-aware Chamfer distance
DEC	embedding clustering
DFS	disease-free survival
DSC	Dice
DINO	*knowledge DIstillation with NO labels*
DNR	Divide-and-Rule
DSC	spatial consistency
DSS	disease-specific survival
E2H	easy to hard
E	esosin
ER	error rate
GAP	global average pooling
H	hematoxylin
HE	hematoxylin and eosin
HR	hazard ratio
iBOT	*image BERT pre-training with Online Tokenizer*
IBS	integrated Brier score

IHC	immunohistochemistry
infoNCE	information noise-contrastive estimation
IOA	interobserver agreement
IOU	intersection over union
K16	Kather 16
K19	Kather 19
κ	Cohen's kappa
KL	Kullback-Leibler
KM	Kaplan-Meier
KNN	k-nearest neighbors
LLRT	log-likelihood ratio test
LL	partial log-likelihood
LOOCV	leave-one-out cross validation
MIL	multiple instance learning
MLSM	multilabel soft margin
MoCo	*Momentum Contrast*
MSI	microsatellite instable
MVA	moving average
OD	optical density
OS	overall survival
READ	rectal adenocarcinoma
ReLU	rectified linear unit
ResNet	residual network
ROI	region of interest
SCG	self-correlation map generating
SGD	stochastic gradient descent
SimCLR	*a Simple framework for Contrastive Learning of visual Representations*
SOTA	state of the art
SPAMS	*SPArse Modelling Software*
SPKM	spherical K-means
SRA	Self-Rule to Adapt
SRMA	Self-Rule to Multi Adapt
SSL	self-supervised learning
SVD	singular value decomposition
SwAV	*Swapping Assignments between multiple Views*
Tanh	hyperbolic tangent
TBC	tumor border configuration
TB	tumor border
TCGA	the cancer genome atlas
TMA	tissue microarray
TMR	tumor to mucin ratio
t-SNE	t-distributed stochastic neighbor embedding

TSR	tumor to stroma ratio
TTE	time-to-event
UDA	unsupervised domain adaptation
UMAP	uniform manifold approximation and projection
ViT	vision transformer
W-F_1	weighted F_1 score
WSI	whole slide image
WSSS	weakly supervised semantic segmentation

1 Introduction

Tant qu'il y aura du malheur, il y aura de l'inspiration.

Enfants terribles, Été triste
Lucas Taupin

This thesis focuses on the use of computer-based algorithms for histopathology, using colorectal cancer (CRC) as a use case. Histopathology is the field of medical research dedicated to diagnosing and studying diseases through tissue samples. When a patient is diagnosed with CRC through screening, oncologists plan the treatment. That decision-making can include pharmaceutical treatment, radiotherapy, and/or surgery. In the case of surgery, the potentially hazardous tissue specimens are resected from the patient, processed, imaged, digitized, and sent for extended diagnosis. These generated digitized images, called whole slide images (WSIs), are snapshots of the tissues at the microscopic level, which pathologists use to determine the tumor stage and, if necessary, make further decisions on treatment planning. This assessment is performed by the evaluation of observable markers in the tissue sample, such as the depth of invasion, lymphovascular status, or metastasis. When available for multiple patients, these clinical markers are sometimes used to predict group survival, thus improving our understanding of the disease and treatment planning. This field of research is known as survival analysis.

The recent advances in scanning techniques allowed WSIs to reach previously unseen image resolution and quality. Moreover, after years of deployment, institutes accumulated considerable amounts of digitized scans. The availability of such a large set of digitized data is an excellent opportunity for research and computational pathology to understand the evolution and behavior of different diseases. Computational pathology is defined as the use of computer-based models and resources to process and analyze WSIs. These models could be used to assist pathologists in their daily routine by helping them to automatically locate and identify the presence of tumors or predict specific biological

markers.

Handling such a massive amount of data is not a trivial task for computers: processing WSIs can be time-consuming and memory-wise expansive due to their high resolution. However, the reduced availability of annotated data remains the main challenge. Indeed, digitized images do not usually include manual annotations, such as cell- or tissue-type labeling at the WSI-level. The few available information is stored at the patient level, in their medical file, where details such as the type of cancer, the depth of invasion, or the resection location are reported.

In addition, some recent computer-based algorithms, namely supervised machine learning, need large amounts of annotated data to be trained and to learn from WSIs. One solution to overcome the lack of labeled data would be for pathologists to manually annotate WSIs to generate complementary data for supervised machine learning. However, in practice, achieving such a task on a large set of data is tedious and time-demanding and, thus, should be avoided. An alternative to supervised approaches that could overcome the need for labeled data is self-supervised learning (SSL). Indeed, self-supervised models use the data structure itself to learn feature representations and, therefore, do not require access to labeled data. As a result, SSL can extract information from all the available data even though they are unlabeled or partially labeled. It implies that publicly accessible databases encompassing millions of tissues can be harnessed.

SSL works based on a two-step logic. First, it is used to learn tissue representation from WSIs using encoders. Encoders are models that aim to synthesize information: when presented with an image depicting a tissue, the encoder compresses its representation and outputs a set of numerical values. Those values, called embeddings or features, encapsulate multiple tissue information such as size, color, or shape. Second, encoders are used to initialize other models performing additional tasks, called downstream tasks, such as tissue classification or segmentation. Since the source SSL model provides valuable tissue embedding, the downstream architectures typically require less annotation to be trained. Consequently, few labeled data are often sufficient to achieve effective task performances, thus reducing the impact of label sparsity.

Still, training SSL model using histopathological data is challenging. Apart from the already mentioned issue on WSIs size, other aspects can harm the learning of the tissue representations. For instance, WSIs cohorts can include scans from multiple institutes. In theory, heterogeneous data strengthen the feature representation of SSL model by proving diverse tissue examples from different sources. However, in practice, discrepancies in the data can create variations in the encoded tissue distribution that are caused by external factors rather than by the intrinsic features of the data. This discrepancy in the data is known as a domain gap. Ideally, images from two distinct institutes should share information once embedded through the encoder if they depict the same type of tissues. However, if a domain gap exists, the two feature representations will not align,

which can hinder the performance of the classification tasks when used for inter-clinical applications.

Tissue segmentation is another limitation related to the application of SSL to histopathological data. Compared to classification, where weakly-labeled data are sufficient for downstream tasks, a segmentation task requires pixel-wise annotations to produce fine-grained outputs. Because such annotations are not available, WSIs output maps tend to have a coarse tissue resolution. It is especially problematic as clinical markers used for survival analysis typically rely on tissue segmentation.

In this thesis, the objective is to tackle the above-mentioned limitations of histopathology. More precisely, this work focuses on building SSL architectures that learn tissue representation from WSIs with limited access to labeled data and using them for various downstream tasks. We first present a model that takes advantage of WSIs staining information and spatial consistency to learn feature representation. Secondly, we address the problems of heterogeneous data and domain gaps when working with multi-source data. Thirdly, we demonstrate how to solve the problem of coarse tissue segmentation using weakly-labeled data and SSL constraints. Finally, based on the previous work, we generate - in an automated way - clinical markers at the patient level. The predicted metrics then serve an extensive survival analysis, which is performed along with expert annotations to validate the approach. These experiments aim to highlight the advantages of SSL in computational pathology and demonstrate its suitability for clinical applications.

1.1 Roadmap of the Thesis

The thesis is divided into six chapters presenting the current state of the art (SOTA), limitations, proposed methods, and conclusions of self-supervision and survival analysis in histopathology.

Chapter 2 - Background and Prerequisites

This chapter introduces the main theoretical components of the thesis. It provides an overview of basic medical knowledge related to the colon and rectal tract, as well as its associated cancer, used to validate the methodology developed. The digitalization of the resected CRC tumors to create WSIs, which allows histopathological diagnosis and analysis, is then explained. Next, the field of computational pathology is presented. It allows us to take advantage of today's computational power to process, classify, and segment WSIs using various algorithms. Then, a SOTA overview of the field of SSL and its applicability to histology to learn tissue representation is provided. Afterward, survival models, used to predict patient survival and hazards based on the learned tissue

feature, are described. Finally, the public datasets and clinical cohorts used in this thesis are listed and described.

Chapter 3 - Divide-and-Rule

This chapter presents our first contribution dedicated to a general solution for tissue representation learning using SSL and constraints in the feature space. To do so, baselines relying on spatial consistency and clustering to regularize the feature space are introduced. Then, an optimization scheme named Divide-and-Rule (DNR), which takes advantage of the WSIs spatial structure to improve feature embedding, is presented. In addition, a stain-based reconstruction for the autoencoder (hematoxylin and eosin (HE) to RGB) is proposed by exploiting the inner properties of the WSIs. Finally, a novel way to create patient descriptors from classified WSIs and use them for survival analysis is introduced.

Chapter 4 - Self-Rule to Multi Adapt

This chapter addresses the domain gap limitation of data from public cohorts caused by variations in the WSIs acquisition process. A novel SSL method called Self-Rule to Multi Adapt (SRMA) is proposed to align feature representation from multi-source datasets to the target space defined by our in-house data. By doing so, all publicly available data can be used to transfer knowledge to in-house (private) cohorts.

Chapter 5 - Coarse to Refined

All the methods presented in the previous chapters output a coarse classification, which hinders the prediction of reliable metrics for diagnosis. In this chapter, the coarse to refined (C2R) approach, which aims to use SSL to refine classification outputs, is introduced. The model takes as input weakly-labeled data, thus alleviating the need for pixel-wise annotations. In addition, a unique solution to validate the models is proposed, using consecutive tissue cuts and specific tissue staining.

Chapter 6 - Building Clinically Relevant Metrics

Throughout this thesis, we focused on tissue classification and segmentation. We can now produce fine-grained tissue maps from WSIs using SSL-based approaches. In this chapter, we propose to take advantage of our previous work to automatize well-established metrics. We then compare our automated approach with existing clinical reports using survival analysis. By doing so, we aim to create a link between research and practical applications.

Chapter 7 - Conclusion

Finally, a review of the main contributions is provided. In addition, the challenges faced and their respective solutions are enumerated. Future directions to improve tissue representation, WSIs classification, and survival analysis are then presented.

1.2 Contributions

In this thesis, we discuss our main research contributions:

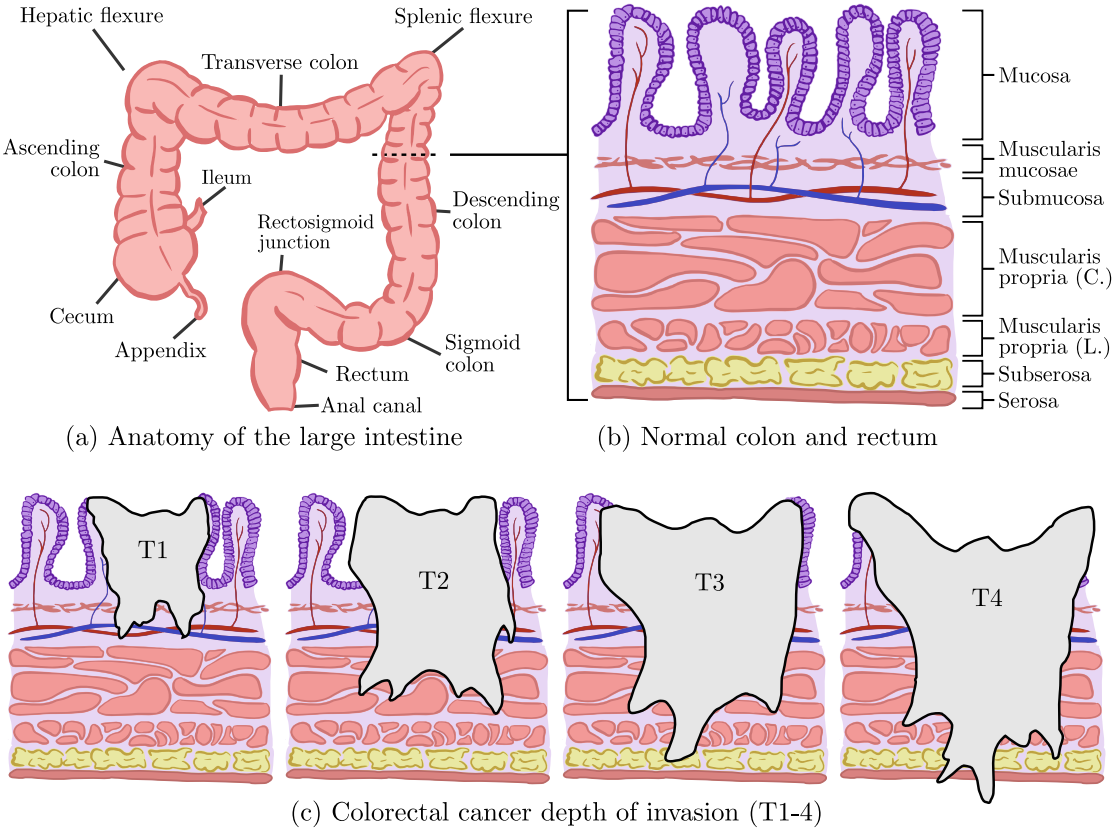
1. Taking advantage of WSIs spatial structure and staining information to learn feature representation in a SSL fashion [4],
2. Proposing a combination of SSL and unsupervised domain adaptation to allow the use of multi-source data from publicly available cohorts [1, 2],
3. Refining segmentation maps by imposing various SSL constraints without the need for fine-grained annotations,
4. Building an automated pipeline to predict clinically relevant metrics as tumor to stroma ratio (TSR) and tumor border configuration (TBC) [3].

2 Background and Prerequisites

> Quand j'ai demandé des sources, on m'a dit "C'est l'homme qui a vu l'homme qui a vu l'ours"... Finalement, c'est peut-être les épines qui ont des roses.
>
> *Storyteller, Storyteller*
> Médine Zaouiche

This chapter explains the scientific background needed to understand the content of the thesis. We explore the main concepts of our research point by point. By the end of this chapter, we aim to give the reader an understanding of the current research and limitations in computational pathology.

To this end, we first provide an overview of the colon (and rectum) structure as well as its associated cancer in section 2.1. We then explain in section 2.2 how digitized images are generated from resected specimens such that they can be used for further diagnosis. In addition, we create a link to the previous section and show how colorectal tissues look from a histological point of view. In section 2.3, we explore the use of computational power to process and learn from the newly acquired images. More specifically, we highlight and discuss the constraints of the three most common learning approaches in computational pathology: supervised, weakly supervised, and self-supervised. In section 2.4, we reach the core concept of our research and go more in-depth about the advantages and uses of self-supervision in computational pathology. In order to take advantage of the learned tissue representation through self-supervision, we examine the field of survival analysis in section 2.5, which is a critical end-task in personalized medicine. Moreover, we take the opportunity to give insights on the public and private data used throughout this research in section 2.6 for both feature representation and survival analysis. Finally, we conclude this chapter in section 2.7.



(a) Anatomy of the large intestine

(b) Normal colon and rectum

(c) Colorectal cancer depth of invasion (T1-4)

Figure 2.1 – Representation of normal colon and tumor depth of invasion. (a) Overall anatomy and subsites. (b) Normal colon layers [105]. (c) Tumor depth of invasion T1-4.

2.1 Colorectal Cancer

Colorectal cancer (CRC) is named after the colon and rectum area. We speak about CRC when the primary tumor site is located in the lower gastrointestinal tract. Worldwide, around two million people are diagnosed with CRC every year, which makes it the third most common cancer [20]. Common risk factors include age (less occurrence in young adults), obesity, alcohol consumption, or family history (first-degree relative previously developed CRC) [24]. Overall, the five-year survival rate is at 68%. However, if the cancer is diagnosed at an early stage, the survival rate goes up to 90%. As a result, the number of CRC related deaths significantly decreased over the years with the development of better screening strategies.

In Switzerland, the statistics on CRC show the same trends. CRC is the fourth most common cancer, with more than 4,500 new cases in 2022, and the second cause of cancer-related mortality [92]. The five-year survival rate is 67% [92].

Before jumping into the specifics of the CRC, we first introduce the anatomical aspect of the normal colorectal area in Figure 2.1a-b. The colorectal tract (or large intestine) is

Table 2.1 – Simplified UICC TNM histopathological classification. T, N, and M indicate depth of invasion, positive lymph node assessment, and presence of distant metastasis, respectively. The prefix p is used to indicate that the variables are validated pathologically.

Staging	pT	pN	pM
I	1-2	0	0
II	3-4	0	0
III	any	1-2	0
IV	any	any	1

composed of the following segments: cecum, ascending colon, hepatic flexure, transverse colon, splenic flexure, descending colon, sigmoid colon, rectosigmoid junction, and rectum [6]. All along the tract, we find regional lymph nodes connected to the lymphatic system as well as tissue irrigation with veins and arteries.

When analyzing a cross-section of the colorectal tract, we can distinguish multiple tissue layers. The inner lining is defined as the mucosa. It is composed of intestinal glands (also called colonic crypts) covered by a single layer of epithelial cells with various functions, such as mucus secretion or water absorption. Attached to it is a thin muscle layer (muscularis mucosae) that separates the mucosa from the submucosa. The submucosa is filled by lymphatic vessels and blood vessels irrigating the inner colon. Further on, we can find two muscle layers, circular and longitudinal muscularis propria, while the outer layers are composed of the subserosa, which contains the fat cells and the serosa. Note that both muscle layers are not always present/visible depending on the resection location.

For cancer staging, we widely refer to the Union for International Cancer Control (UICC) TNM classification system that grades cancer status based on T, N, and M categories. The simplified classification process is depicted in Table 2.1. T refers to the size and depth of the main tumor, namely how deep the cancer has grown in the organ. N is linked to the number of positive regional lymph nodes, and M is the presence of metastasis (spread of the main tumor to distant organs). We refer as pT, pN, and pM the variables T, N, and M that have been validated post-surgery at the histopathological level [120].

As illustrated in Figure 2.1c, the pT stage in CRC is linked to the depth of invasion where pT1, pT2, pT3, and pT4 refers to the progression of the main tumor into the submucosa, muscuaris propria, subserosa/adventitia, and serosa, respectively. Lesions limited to the normal mucosa (*i.e.* which do not go through the muscularis mucosae) are referred to as *in situ* (Tis). In addition, the pN grading assesses the spread to regional lymph nodes. It is defined as no spread to regional lymph nodes (pN0), up to three infiltrated lymph nodes (pN1), and more than three infiltrated lymph nodes (pN2). Finally, the presence of metastases is labeled as pM1. For CRC, the most common site of metastases is the liver.

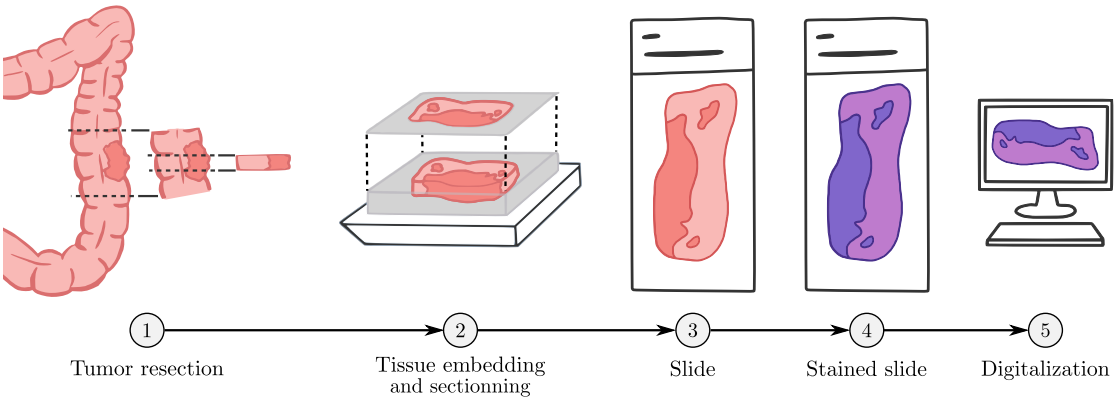


Figure 2.2 – Digitization pipeline of a resected specimen. (1) Resection of the specimen and selection of a representative area. (2) Fixing, embedding, and sectioning of the specimen. (3) Resulting slide. (4) Staining of the slide using HE. (5) Final digitized whole slide image (WSI).

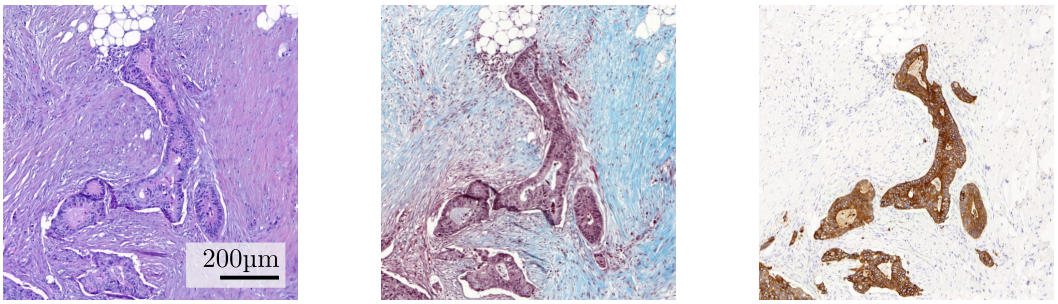
Note that more detailed subcategories exist for cancer grading. They are labeled using letters (*e.g.* pN1c) but are not covered in this document. In addition to the mentioned T, N, and M variables, other metrics are assessed in clinical reports, such as venous invasion, tumor grade, or budding. For an extended definition of clinical variables, please refer to the supplementary material in section A.3.

2.2 Specimen & Slide Preparation

The diagnosis of CRC through screening or biopsies is typically followed by the resection of the main tumor. After resection, the tumor is processed, fixed, analyzed, and then graded by pathologists. The analysis of the specimen can be done through a microscope or using a computer by visualizing a digitized version of the specimen. Digital pathology is hence defined as the digitization process and data management of specimen slides.

In this section, we describe the process for tissue preparations composed of a series of consecutive steps [61, 93]. The overall procedure is depicted in Figure 2.2. First, one to multiple tissue samples are selected from representative areas of the resected specimen. The so-called representative areas are small tissue samples that might contain relevant information for diagnosis. It includes, for example, the center of the primary tumor, tumor front, or regional lymph nodes (Figure 2.2-1).

The tissue is then fixed using formaldehyde (also known as formalin when dissolved in water). The primary purpose of fixation is to preserve tissue by retaining its morphological and chemical characteristics as much as possible. When applied to the tissue, the formalin solution shows a penetration rate of approximately 1 mm an hour. As a result, large specimens require more extended fixation periods than biopsies, which can usually be



(a) HE (b) Trichrome (Fast Green + Fuchsin + Hema.) (c) IHC (DAB + Hema.)

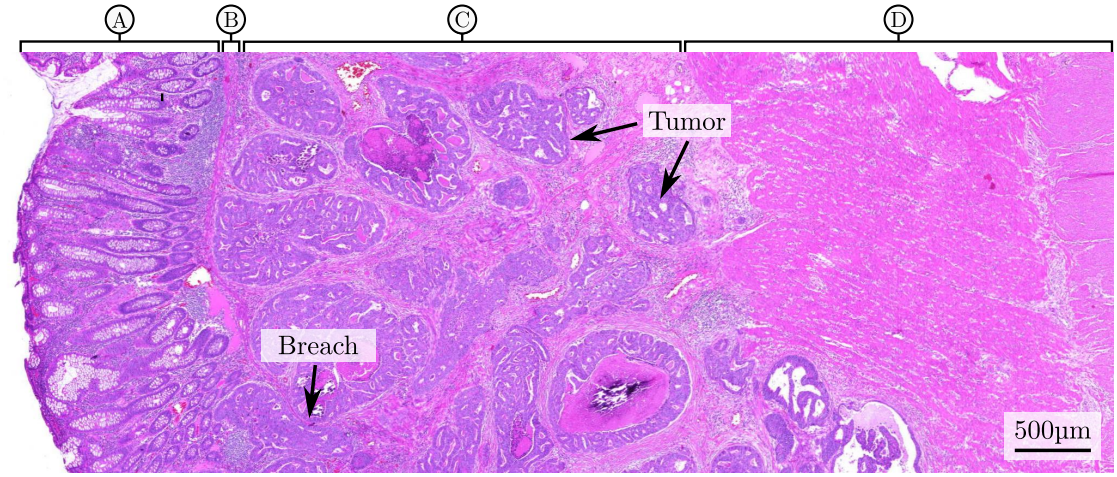
Figure 2.3 – Staining of three consecutive cuts using hematoxylin and eosin (HE), trichrome, and immunohistochemistry (IHC). (a) Hematoxylin (purple) and eosin (pink) staining. (b) Trichrome staining with fast green (blue/green), fuchsin (red), and hematoxylin (purple). (c) IHC with hematoxylin (purple) and DAB (brown).

processed in a single day.

After fixation, the tissue goes through dehydration, whose role is to remove water and fixative and to further harden the tissue. Next, the tissue is embedded in liquid paraffin. Here, the orientation of the tissue is critical as it will determine the cutting axis. For example, colon tissue needs to be oriented such that, after cutting, we highlight all the tissue layers (*i.e.* perpendicular to the inner colon surface). Finally, the tissue is sectioned using a microtome (Figure 2.2-2). The resulting slices are typically between $2-5$µm thick, so a single layer of cells is visible. Note that an alternative for tissue preparation is named frozen section. This approach generates a poor-quality output with major artifacts but has the advantage of being extremely fast. The tissue can typically be processed in a few minutes and is mainly used when urgent diagnoses are needed (*e.g.* for intraoperative tumor resection).

The slides are now fixed and technically ready for visualization (Figure 2.2-3). However, in the current setting, the samples show poor contrast between elements and are hardly interpretable. As a result, the slices need to be further stained. The staining is a chemical process that artificially highlights/enhances tissue components using dyes. (Figure 2.2-4). Several types of staining can be applied depending on the desired features that need investigation. The most common staining is hematoxylin and eosin (HE). The dyes react based on the basic and acidic composition of the tissue. The hematoxylin stains nuclei of cells in purple, while the eosin stains the cytoplasm, extracellular matrix, and collagen in pink. Another practice is Masson's Trichrome staining, which enhances the presence of collagen fibers. It is helpful to identify stroma and muscle tissue as they can be hardly distinguishable using HE images when no contextual information is available. Finally, we cite immunohistochemistry (IHC), which aims at targeting specific proteins. It comes in handy when there is a need to identify small patterns, such as isolated tumor cells.

(a) Normal colon



(b) Tumor invasion of tissue

Ⓐ Normal mucosa Ⓑ Muscularis mucosae Ⓒ Submucosa Ⓓ Muscularis propria Ⓔ Subserosa/serosa

Figure 2.4 – Visualization of the colon tissue layers after scanning. (a) Normal colon. (b) Invasion of the tumor through the submucosa to the muscularis propria.

To visualize the differences in staining appearances, we show in Figure 2.3 an example of three consecutive cuts stained with different approaches. We assume we have one-to-one feature correspondence between the three cuts as their cutting planes are, by definition, a few micrometers apart. The first image depicts an HE WSI region with a tumor cells cluster structure in the center. In the second image, we observe the result of a Masson's Trichrome stain that allows us to differentiate between collagen fibers in blue, cytoplasm in red, and cell nuclei in purple. Finally, on the far right section, the IHC stain where antibodies (*i.e.* AE1 and AE3) are selected to highlight cancer that forms in epithelial tissue (*i.e.* carcinoma) in dark brown. Note that the AE1 and AE3 antibodies target epithelial cells in general. As a consequence, normal mucosa will also appear brown.

Once stained, the images are scanned using a high-resolution camera (*e.g.* 0.25µm/pixel)

to create the final WSI (Figure 2.2-5). Knowing that microscope glass slides are typically 25mm × 75mm , a single digitized image can reach 0.1megapixels × 0.3megapixels, corresponding to a total of 30 gigapixels. Consequently, WSIs are often called gigapixel images. We often deal with the output magnification when working with WSIs. It defines the objective lens employed by the scanner and is independent of the camera's resolution [127]. Still, both values are inversely proportional. Given an output resolution of the tissue at 20× (*e.g.* 0.5µm/pixel), we can expect its resolution to halve if we use a magnification at 10× (*i.e.* 1µm/pixel). The same logic applies when doubling the magnification to 40× (*i.e.* 0.25µm/pixel).

A visualization of the generated output from a normal colon sample is depicted in Figure 2.4. In the first row, we highlight the main normal tissue layers as described in the previous section. On the far left, the normal mucosa can easily be identified by its colon crypts that are "flower shaped". It is followed by the muscularis mucosae, which acts as a boundary between the normal mucosa and the submucosa supplying it. The muscularis propria appears to the right of the submucosa with both circular and longitudinal muscle layers. Finally, on the far right, we observe the subserosa and serosa that form the limit of the organ. On the bottom row, we can observe a tumor infiltration of the normal tissue. Following the muscularis mucosae, we can identify the region where the tumor breached through the boundary, allowing itself to progress further into the submucosa and muscularis propria.

2.3 Computational Pathology

So far, we have described how to acquire digitized images. It is now time to go further and learn how to take advantage of the generated data to perform clinical analysis. In daily diagnosis, pathologists use either microscopes or visualization software to review cases and perform tumor grading. These dedicated software often include various automated tools to process the visualized image.

Computational pathology, sometimes called CPATH, uses computational power to process histopathological images. It is often regarded as a complementary field that aims to help pathologists in decision-making. Computational pathology has the advantage of being able to process, learn, and synthesize large amounts of data. Various tools have been developed to perform basic tasks such as stain estimation and extraction, tissue description, or cell classification. For example, we recommend using QuPath [11], which is an open software that allows simple visualization and processing of WSIs.

In this section, we first introduce the basics of WSIs stain extraction (subsection 2.3.1) and normalization (subsection 2.3.2). Next, we explain how we can take advantage of computational resources and machine learning to achieve various end tasks such as feature representation or tissue classification (subsection 2.3.3). Finally, we give an overview of

the metrics used in this document to assess the quality of the feature representations (subsection 2.3.4).

2.3.1 Stain Extraction

WSIs use various stains such as HE to highlight tissue features. Nevertheless, the appearance of stains can vary between slides based on their tissue density or the scanning device. To solve this, we use stain extraction to isolate dye components from WSIs for further normalization.

Let's assume we have an RGB representation $\mathbf{x} \in [0,1]^3$ of a pixel. Here, we want to estimate its corresponding staining density $\mathbf{s} \in \mathbb{R}_+^{N_s}$ where N_s is the number of stains (*e.g.* $N_s = 2$ for HE). Light transmission depends on stains' concentration in a non-linear way [122]. Therefore, doubling the pixel values does not mean doubling its visual intensity. Hence, we define the optical density (OD) equivalent of the input:

$$\mathbf{x}' = \frac{\log(\max(\mathbf{x}, \epsilon))}{\log(\epsilon)}, \tag{2.1}$$

where $\epsilon \ll 1$ is used to fix numerical instability when dealing with extreme values. The computed OD is linearly proportional to the optical concentration of the stain. We then define a conversion matrix $\mathbf{V} \in \mathbb{R}^{N_s \times 3}$ and its inverse $\mathbf{V}^{-1} \in \mathbb{R}^{3 \times N_s}$ to move from the color space to the staining space and vice-versa. The relation between the RGB values and staining representation is:

$$\mathbf{s} = \mathbf{x}'\mathbf{V}^{-1} \qquad \text{and} \qquad \mathbf{x}' = \mathbf{s}\mathbf{V}. \tag{2.2}$$

If the number of stains is lower than the one of RGB channels (*i.e.* $N_s < 3$), the inverse of the conversion matrix is not defined. To overcome this issue, we use the Moore-Penrose pseudo inverse $\mathbf{V}^\dagger$:

$$\mathbf{V}^\dagger = \mathbf{V}^\top(\mathbf{V}\mathbf{V}^\top)^{-1}. \tag{2.3}$$

Most staining methods typically include two to three stains per WSI. The design and creation of the conversion matrix $\mathbf{V}$ is essential to move from the RGB to the staining space. We present the Ruifrok [122], Macenko [96], and Vahadane [139] methods, three well-known strategies to estimate it. For better understanding, a visual comparison between the approaches on a HE tile is depicted in Figure 2.5.

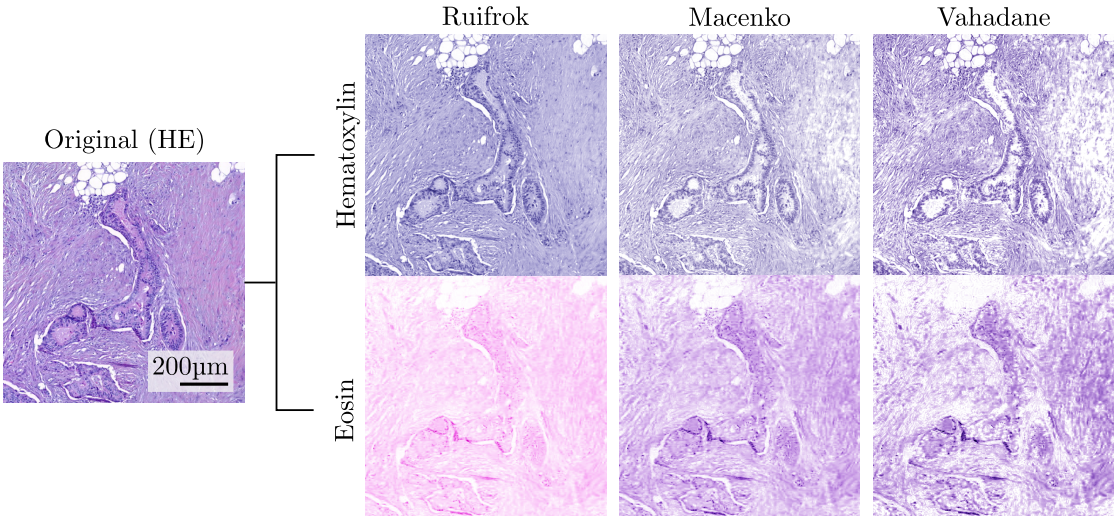


Figure 2.5 – Example of stain estimation from an RGB image into its hematoxylin and eosin (HE) components. We compare the three main approaches: Ruifrok [122], Macenko [96], and Vahadane [139].

Ruifrok Estimation

In Ruifrok [122], they use pure staining concentration to estimate the conversion matrix. In other words, they use single stained WSIs to measure absorption for red, green, and blue channels. The results are aggregated over multiple regions to get the estimated conversion matrix for hematoxylin (H), esosin (E), and diaminobenzidine (DAB):

$$\mathbf{V} = \begin{matrix} \\ \text{H} \\ \text{E} \\ \text{DAB} \end{matrix} \begin{matrix} \begin{matrix} \text{R} & \text{G} & \text{B} \end{matrix} \\ \begin{pmatrix} 0.18 & 0.20 & 0.08 \\ 0.01 & 0.13 & 0.01 \\ 0.10 & 0.21 & 0.29 \end{pmatrix} \end{matrix} . \tag{2.4}$$

H, E are the two channels that form HE stained images. DAB highlights the content of IHC slides. Together, these three components represent the most common stains in histology. The matrix is then normalized row-wise to get the final estimation of the conversion matrices:

$$\mathbf{V}_{\text{ruifrok}} = \begin{pmatrix} 0.65 & 0.70 & 0.29 \\ 0.07 & 0.99 & 0.11 \\ 0.27 & 0.57 & 0.78 \end{pmatrix} \quad \text{and} \quad \mathbf{V}^{-1}_{\text{ruifrok}} = \begin{pmatrix} 1.88 & -1.02 & -0.55 \\ -0.07 & 1.13 & -0.13 \\ -0.60 & -0.48 & 1.57 \end{pmatrix} . \tag{2.5}$$

The matrices are fixed and can directly be applied to predict stain concentration in WSIs.

A database of other stain estimations using the same approach is available online [87].

Macenko Estimation

When dealing with data from different institutes, we can see the limitations of using a fixed conversion matrix. The estimation quality is directly affected by various factors, such as the tissue's local thickness or scanner settings.

In this context, an approach that can compute robust statistics from individual WSIs is more appropriate, such as Macenko [96] that uses singular value decomposition (SVD) to extract and isolate staining components. More formally, let $\mathbf{M} \in \mathbb{R}^{N\times 3}$ be a set of N OD measurements of RGB pixels from a WSI (Equation 2.1). We perform SVD on the measurements' covariance matrix to extract the eigenvectors. As the covariance matrix is real symmetric, the SVD computation is relaxed and becomes:

$$\operatorname{cov}(\mathbf{M},\mathbf{M}) = \mathbb{E}[(\mathbf{M}-\mathbb{E}[\mathbf{M}])(\mathbf{M}-\mathbb{E}[\mathbf{M}])^\top] \overset{\text{SVD}}{=} \mathbf{U}^\top \Lambda \mathbf{U}, \tag{2.6}$$

where $\mathbb{E}$ is the expectation operator, $\mathbf{U}$ contains the eigenvectors of the covariance matrix, and Λ is a diagonal matrix with eigenvalues as entries. We assume that the two largest eigenvalues are enough to capture most of the staining information. We then project the measurements on the new 2D basis formed by the eigenvector and compute the resulting angles $\phi \in \mathbb{R}^N$ for each entry as:

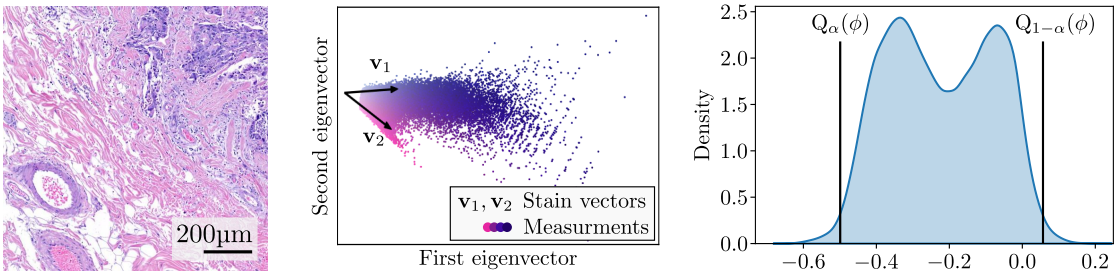
$$\phi = \arctan2\left(\mathbf{MU}\right), \tag{2.7}$$

where arctan2 is a function that measures the angles of the resulting vectors with respect to the origin. Figure 2.6 shows an example of angle distribution across the measurements. Based on the distribution of the angle, we select the top α-th and $(1-\alpha)$-th percentiles to get an estimation of the support vectors that are less sensitive to outliers as:

$$\phi_{\alpha_{\text{th}}} = \mathrm{Q}_\alpha(\phi) \qquad \text{and} \qquad \phi_{(1-\alpha)_{\text{th}}} = \mathrm{Q}_{(1-\alpha)}(\phi). \tag{2.8}$$

Here Q_α denotes the quantile function for a given $\alpha \in [0,1]$. Based on the newly computed angles, we can retrieve their support vectors:

$$\mathbf{v}_1 = \mathbf{U}\begin{pmatrix}\cos(\phi_{\alpha_{\text{th}}})\\ \sin(\phi_{\alpha_{\text{th}}})\end{pmatrix} \qquad \text{and} \qquad \mathbf{v}_2 = \mathbf{U}\begin{pmatrix}\cos(\phi_{(1-\alpha)_{\text{th}}})\\ \sin(\phi_{(1-\alpha)_{\text{th}}})\end{pmatrix}. \tag{2.9}$$



(a) Input sample (b) Projection on eigenvectors (c) Distribution of angles ϕ

Figure 2.6 – Stain estimation using the Macenko approach. (a) Input RGB measurements. (b) 2D projection of the measurements based on matrix decomposition and estimated stains. (c) Distribution of angles ϕ based on the two largest eigenvectors as well as the estimated quantiles Q_α and $Q_{1-\alpha}$ (α=0.99).

In practice, we use $\alpha = 0.99$. The newly estimated conversion matrix is defined as the concatenation of both support vectors:

$$\mathbf{V}_{\text{macenko}} = \begin{pmatrix} \mathbf{v}_1 & \mathbf{v}_2 \end{pmatrix}^\top. \tag{2.10}$$

It is also important to mention that discarding measurements with high luminosity improves the quality of the estimation. Such values are considered part of the background and do not carry any stain information. It is done by converting the image to the LAB colorspace [28] and applying a threshold $\delta_l \in [0, 1]$ on the luminescence channel. The threshold is typically high and set to $\delta_l = 0.8$.

Vahadane Estimation

In Vahadane [139], they propose a stain separation approach that maximizes the representation sparsity and assumes a non-negative stain density. Here, we start with the same set of OD measurements $\mathbf{M} \in \mathbb{R}^{N\times 3}$ from a WSI. We then use dictionary learning to find a positive matrix $\mathbf{V} \in \mathbb{R}_+^{N_s\times 3}$, where N_s is the size of the dictionary (*i.e.* number of stains) that satisfies:

$$\arg\min_{\mathbf{V},\mathbf{D}} \|\mathbf{M} - \mathbf{D}\mathbf{V}\|_2^2 + \lambda \sum_{i=1}^{N} \|\mathbf{d}_i\|_1, \tag{2.11}$$

$$\text{where} \quad \|\mathbf{v}_j\|_2^2 \leq 1, \forall j \in \{1, \ldots, N_s\}.$$

The values of $\mathbf{D} = \begin{pmatrix} \mathbf{d}_0 & \cdots & \mathbf{d}_N \end{pmatrix} \in \mathbb{R}_+^{N\times N_s}$ are jointly optimized to ensure sparsity of the representation through the l_1-norm. Dictionary learning will not be discussed in

this work. However, such optimization constraints are well known, and we recommend referring to *SPArse Modelling Software* (SPAMS) [97] for additional information and implementation.

2.3.2 Stain Normalization

One of the main advantages of dynamic stain estimation is the normalization of the staining information between WSIs. Basically, given a target and a source WSI, we can normalize the entries of the source image such that it follows the color distribution of the target one. Such a technique comes in handy when using models trained on cohorts whose staining concentration and appearance differ from in-house data.

More formally, let's assume two sets of OD measurements $\mathbf{M}_{\text{src}} \in \mathbb{R}^{N\times 3}$ and $\mathbf{M}_{\text{tar}} \in \mathbb{R}^{M\times 3}$ which represent our source and target image, respectively. We extract their stain concentration using one of the previously defined approaches (*e.g.* Macenko) to get $\mathbf{V}_{\text{src}}, \mathbf{V}_{\text{tar}} \in \mathbb{R}^{N_s\times 3}$. The staining representations $\mathbf{S}_{\text{src}} \in \mathbb{R}^{N\times N_s}$ and $\mathbf{S}_{\text{tar}} \in \mathbb{R}^{M\times N_s}$ are given by:

$$\mathbf{S}_{\text{src}} = \mathbf{M}_{\text{src}}\mathbf{V}_{\text{src}}^{-1} \qquad \text{and} \qquad \mathbf{S}_{\text{tar}} = \mathbf{M}_{\text{tar}}\mathbf{V}_{\text{tar}}^{-1}. \tag{2.12}$$

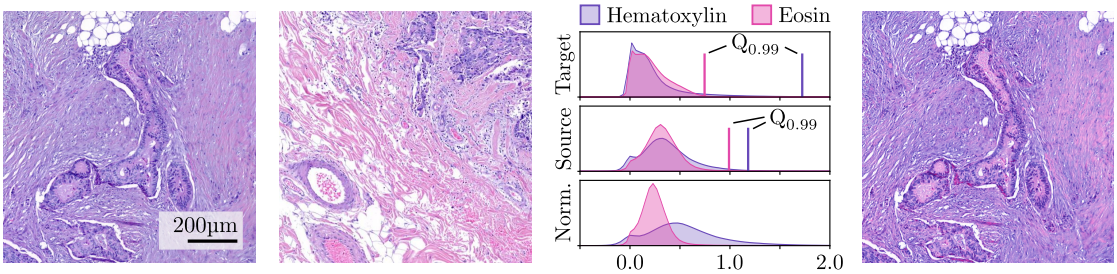
For each representation, we extract the range of stain concentrations. For the lower bound, we assume that the staining representation cannot be negative (*i.e.* $(\mathbf{S})_{i,j} \geq 0, \forall i,j$). For the upper bound, we use the $(1-\alpha)$-th quantile of the source and target stain representation to be less sensitive to outliers. We define the normalized stain as the re-scaled source concentration:

$$\mathbf{S}_{\text{norm}} = \begin{pmatrix}\mathbf{S}^i_{\text{norm}} & \cdots & \mathbf{S}^{N_s}_{\text{norm}}\end{pmatrix} \qquad \text{and} \qquad \mathbf{S}^i_{\text{norm}} = \frac{\mathrm{Q}_{(1-\alpha)}(\mathbf{S}^i_{\text{tar}})}{\mathrm{Q}_{(1-\alpha)}(\mathbf{S}^i_{\text{src}})}\mathbf{S}^i_{\text{src}}, \tag{2.13}$$

where $\mathbf{S}^i$ is the i-th stain entry and $i \in \{1,\ldots,N_s\}$. Note that a commonly used value for upper bound estimation is $\alpha = 0.99$. Finally, the normalized concentration is projected back to the OD RGB space as:

$$\mathbf{M}_{\text{norm}} = \mathbf{S}_{\text{norm}}\mathbf{V}. \tag{2.14}$$

An example of stain normalization is given in Figure 2.7, where we can observe the normalization of a source image to the target color distribution. We also display the detected stain distributions for the source, target, and normalized image. Other methods have been developed to solve the problem of image normalization, such as the use of the



(a) Source (b) Target (c) Stain densities (d) Normalized

Figure 2.7 – Stain normalization using the Macenko approach. (a) Source image to normalize. (b) Target distribution to match. (c) Distribution of stains for source and target image as well as their estimated quantiles Q_α ($\alpha = 0.99$) and the resulting normalized stain concentration. (d) Normalized source image based on target stain.

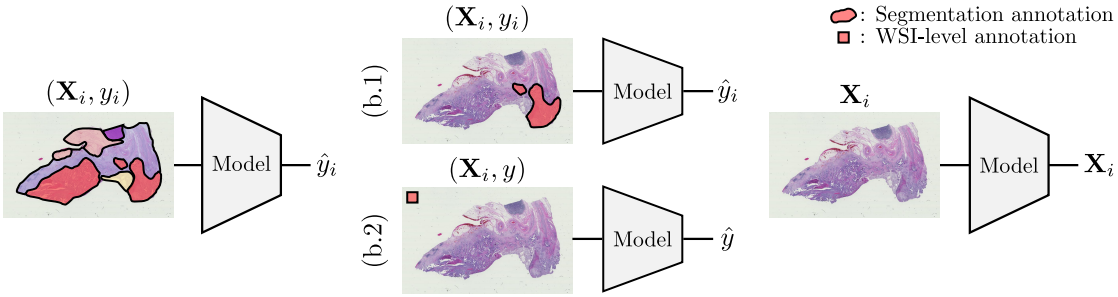
LAB colorspace instead of the conventional RGB space [134], or more elaborated methods that take advantage of neural networks to train Gaussian mixture models [156]. However, complex methods tend to increase computational time while only slightly improving the result.

Before concluding, we list good practice recommendations to improve the quality of image normalization. First, to avoid negative stain concentration, we encourage the use of regularized optimization instead of pseudo inverse to impose sparsity and non-negativity [139]. Secondly, we recommend using simple tricks to reduce the computational time [7]. Finally, it is essential to note that the normalization performance is linked to the image's color statistics. As a result, we suggest using large areas that include heterogeneous tissue representations to cover a wide range of stain intensities.

2.3.3 Learning Feature Representation

When it comes to computational pathology, we rely on large WSIs that embed rich multi-level tissue features (*i.e.* tissue available at different magnifications). A question then arises: "How can we take advantage of such a large amount of data for clinical applications?". The answer to that question is not trivial, as it mainly depends on the type of data available. For example, a straightforward solution would be to build handcrafted features and use them to perform downstream tasks. However, such a solution comes with two main drawbacks. First, it is time-consuming as the development of handcrafted features often relies on a try-and-error strategy, and second, it is based on the assumption that we have prior knowledge of what could be sets of discriminant features for WSIs representation.

Here, we propose to use machine learning and neural networks to extract information from WSI. Neural networks have the advantage of being able to learn discriminant information by themselves. Moreover, machine learning can be used for various end



(a) Supervised (b) Weakly supervised (c) Self-supervised

Figure 2.8 – Presentation of the three main learning approaches for feature representation as (a) supervised, (b) weakly supervised, and (c) self-supervised. Each WSI is composed of multiple crops. We denote the patch, patch labels, WSI-level label, patch predictions, and WSI-level prediction as $\mathbf{X}_i$, y_i, y, $\hat{y}_i$, and $\hat{y}$, respectively. The color overlays indicate the location of labeled tissue classes.

tasks such as tissue classification, segmentation, or survival prediction. We list the three most common learning strategies used in computational pathology to process and learn from large histopathological datasets. A representation of the mentioned approaches is depicted in Figure 2.8.

Supervised Learning

Let's assume we have access to a WSI named $\mathcal{W}$. This WSI is a composition of smaller RGB patches that we denote as $\mathbf{X}_i \in \mathcal{W}$, where $i \in \{1, 2, \ldots, |N|\}$ is the index of the image within the WSI. In the case of supervised learning, for each image tile, we have access to a label y_i that indicates the type of tissue present in the image such that it forms a pair $(\mathbf{X}_i, y_i)$. The pair is used to train a model whose output $\hat{y}_i$ is compared to the original label. Based on the predictions, the weights of the architecture are then optimized until convergence (Figure 2.8a).

This procedure works well on large datasets such as ImageNet [42], where full annotations are available. However, this is scarcely true in histopathology. Due to the large size of WSIs, it is highly time-consuming to fully label them. Moreover, when it comes to segmentation tasks, creating masks is sometimes unfeasible as certain areas are composed of a mixture of tissues whose delimitations are barely distinguishable. The procedure often requires additional staining to identify the presence of specific markers and tissues.

Weakly-supervised Learning

We refer to weakly-supervised learning when we rely on limited or imprecise labeled data, which arises in different scenarios.

The first example is partially annotated WSI where only subareas of the WSI are labeled (Figure 2.8b.1). In this case, an expert annotator explicitly locates the presence of specific tissue regions (*e.g.* tumor) within the WSI. We endup with few labeled pairs $(\mathbf{X}_i, y_i)$ with $i \in \{1, 2, \cdots, M\}, M \ll N$. Knowing the local labels, it is still possible to follow a standard supervised learning approach to train our network. However, the selected areas only represent a small part of the WSI, and as we lack additional information about the class distribution of the unlabeled areas, the remaining areas are usually discarded.

A second example is WSI-level labels (Figure 2.8b.2). In this case, we have access to a single label y for each WSI. WSI-level labels are often available after clinical reports, alongside various information such as the WSI staining, the presence of tumor within the slide, or the type of organ it comes from. Such information can be processed to learn discriminant features. However, one of the main drawbacks of this approach is the localization of the region of interest. For example, when a WSI is labeled as "tumor", it does not necessarily mean that the whole tissue is composed of tumoral tissue but at least part of it. As a result, the designed architecture often needs to implement a ranking system to retrieve the positive tiles at inference time. Moreover, learning from gigapixel images using a single label is difficult as it can quickly converge to trivial solutions.

Self-supervised Learning

Last but not least is the use of self-supervised learning (SSL). In this setup, we do not have access to any of the labels. As a result, we must rely on something other than the usual supervised methods to train the model. Here, the goal is to learn relevant feature discriminators for future downstream tasks. Self-supervised approaches try to create their own supervision from the input data (Figure 2.8c) to learn feature representations. SSL is often mixed with unsupervised learning as both fields assume missing labels. In fact, SSL is a branch of unsupervised learning. We speak about SSL whenever the model creates its own supervision from the data. On the other hand, unsupervised learning define all methods that do not rely on labeled data, such as clustering, anomaly detection, or SSL [10].

Let's assume we have access to a large amount of data from clinical WSIs and that we are able to learn in a self-supervised fashion a feature extraction model such that $\mathbf{z}_i = f_\phi(\mathbf{X}_i)$, where f_ϕ is the model with learned parameters ϕ and $\mathbf{z}_i$ the corresponding extracted features from a tissue sample $\mathbf{X}_i$. In addition, we assume that the model is designed such that the dimensionality of the feature space is much lower than the one of the input image. It means that for every tissue sample, we can get a compressed feature representation of the input, thus easing future downstream tasks. Such models are often called pre-trained or feature extractor models.

The use of pre-trained models comes with multiple advantages. Firstly, as they do not

need labels, they can be trained on full cohorts to capture a wide range of tissue features. This is critical as clinical cohorts are mainly composed of unlabeled data. Secondly, as the model is trained on large cohorts, it is not task-specific, meaning that the same pre-trained model can be used to solve both classification and segmentation problems. Lastly, as the pre-trained models provide a compressed representation of tissues, it typically lowers the number of annotations needed for downstream tasks. It is interesting as it saves experts precious time.

2.3.4 Metrics

We finish this section by reviewing the principal metrics used in this research to assess the quality of the classification and segmentation predictions.

F_1-score

For classification, we prefer the use of F_1-score as it considers the distributions and occurrences of classes. Given a set of binary labels and predictions, we define the class-wise F_1-score:

$$P = \frac{TP}{TP + FP}, \qquad R = \frac{TP}{TP + FN}, \qquad \text{and} \qquad F_1 = 2\frac{PR}{P + R}, \tag{2.15}$$

where TP, FP, and FN are the number of true positives, false positives, and false negatives, respectively.

When dealing with multiple classes, the F_1-score is computed class-wise and afterward averaged. Three main averaging strategies exist: micro, macro, and weighted. The selection of the method is based on the end task. If the classes have equal importance but different densities, we recommend using the macro score (*i.e.* average of F_1-scores across classes). Otherwise, the weighted solution is preferred (*i.e.* weighting average of F_1-scores based on class densities). The micro score is less common as it is similar to a simple accuracy measure.

Dice and Intersection over Union

When it comes to evaluation segmentation, we typically use Dice (DSC) or intersection over union (IOU) score:

$$\text{DSC} = \frac{2|A \cap B|}{|A| + |B|} \qquad \text{and} \qquad \text{IOU} = \frac{|A \cap B|}{|A \cup B|}, \tag{2.16}$$

where A and B are ground truth and predictions map, respectively. By taking a closer look at the DSC metric, we can see that its definition is the same as the F_1-score. In the literature, DSC is sometimes also referred to as Sørensen–Dice. As for IOU, it is occasionally called Jaccard-Index. We scarcely report both DSC and IOU as they are directly linked.

$$\mathrm{DSC} = 2\frac{\mathrm{IOU}}{1+\mathrm{IOU}}. \tag{2.17}$$

In this work, we prefer the use of DSC score to stay consistent with the evaluation of the classification tasks.

Interobserver Agreement

One aim of this thesis is to provide automated approaches to assess clinical values to lighten pathologist workload and allow analysis of large cohorts. The proposed automated solution requires validation against experts' annotations. To do so, we use the Cohen's kappa (κ) coefficient, also known as interobserver agreement (IOA), which measures the level of agreement between two raters. For the binary case, it is defined as:

$$\mathrm{IOA} = \frac{2(\mathrm{TP}\cdot\mathrm{TN} - \mathrm{FN}\cdot\mathrm{FP})}{(\mathrm{TP}+\mathrm{FP})(\mathrm{FP}+\mathrm{TN}) + (\mathrm{TP}+\mathrm{FN})(\mathrm{FN}+\mathrm{TN})}, \tag{2.18}$$

where TP, TN, FP, and FN are the number of true positives, true negatives, false positives, and false negatives, respectively. For continuous variables, we recommend the use of Pearson correlation or r^2 coefficient of determination.

By ways of comparing, IOA value within range $[0, 0.2]$, $]0.2, 0.4]$, $]0.4, 0.6]$, $]0.6, 0.8]$ and $]0.8, 1.0]$ are referred to as none, fair, moderate, substantial and near perfect agreement respectively [99].

Chamfer Distance

Last but not least is the evaluation of tissue structure segmentation. The DSC score provides the overall performance of a classification method but does not take into account the spatial distribution of the output. Therefore, when detecting small patterns such as tumor cells or invasion, we need to ensure we preserve the object structures. To do so, we report density-aware Chamfer distance (DCD) that evaluates spatial coherence of predictions [150]. More formally, let $S_x, S_y \subset \mathbb{R}^2$ be the sets of points from the labels and predictions maps, respectively. We define DCD as:

Metric	Pred. 1	Pred. 2
DSC	97.6	97.7
DCD	0.024	0.006

Figure 2.9 – Comparison between Dice (DSC) and density-aware Chamfer distance (DCD) metrics based on ground truth and different predictions over two classes A and B. Both outputs achieve a similar DSC score. A higher DCD score highlights a better preservation of the class structures.

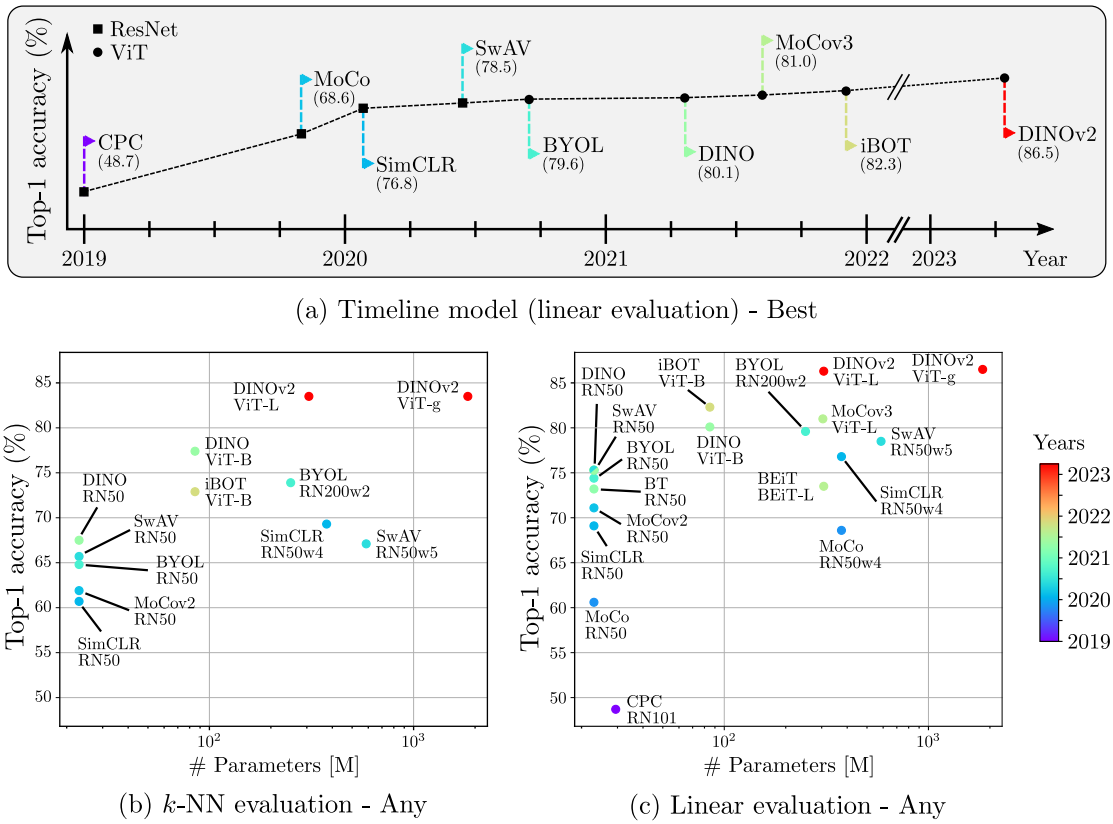
$$\mathrm{DCD} = \frac{1}{2|\mathcal{S}_x|}\left(\sum_{\mathbf{x}\in\mathcal{S}_x} 1 - \frac{1}{n_{\hat{y}}} e^{-\alpha\|\mathbf{x}-\hat{\mathbf{y}}\|_2}\right) + \frac{1}{2|\mathcal{S}_y|}\left(\sum_{\mathbf{y}\in\mathcal{S}_y} 1 - \frac{1}{n_{\hat{x}}} e^{-\alpha\|\mathbf{y}-\hat{\mathbf{x}}\|_2}\right), \tag{2.19}$$

$$\hat{\mathbf{y}} = \min_{\mathbf{y}\in\mathcal{S}_y} \|\mathbf{x}-\mathbf{y}\|_2, \qquad \text{and} \qquad \hat{\mathbf{x}} = \min_{\mathbf{x}\in\mathcal{S}_x} \|\mathbf{x}-\mathbf{y}\|_2,$$

where $n_{\hat{y}}$ and $n_{\hat{x}}$ are local density estimators, and $\alpha \in \mathbb{R}_+$ is a temperature factor. Here, for each point $\mathbf{x} \in \mathcal{S}_x$ in the label set, we compute its distance to the closest point $\hat{\mathbf{y}} \in \mathcal{S}_y$ in the evaluation set. The distance is then weighted based on the local density (*i.e.* total number of points in $\mathcal{S}_x$ sharing the same $\hat{\mathbf{y}}$ as $\mathbf{x}$). We then mirror the operation by considering $\mathcal{S}_y$ as the evaluation set instead of $\mathcal{S}_x$. The final prediction is the average of both metrics, which should be minimized. In our case, we assume the local density is $n_{\hat{x}} = n_{\hat{y}} = 1$ as the local point density is the same in the 2D image plane. In Figure 2.9, we show an example of the structure-preserving aspect of the DCD. While DSC scores are similar, the second prediction shows more spatial consistency with the reference labels and thus achieves much lower DCD.

2.4 Self-supervision and Computational Pathology

When it comes to histopathological data, we have access to a large amount of unlabeled data. As they lack annotations, those data are usually discarded in standard supervised approaches. In this section, we introduce SSL that aims to learn feature embedding from unlabeled data. Such optimization models are able to build strong feature encoders that can later be used for downstream tasks. Recent works use encoders' architectures such as residual networks (ResNets) [65] or vision transformers (ViTs) [47] to embed features. However, in most cases, the encoder selection is independent of the formulation of the problem.



(a) Timeline model (linear evaluation) - Best

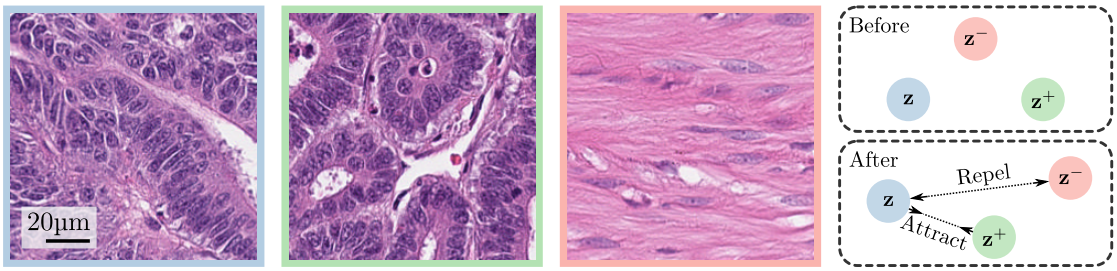
(b) k-NN evaluation - Any

(c) Linear evaluation - Any

Figure 2.10 – Evolution of the top performing SSL models and backbones on ImageNet-1K classification. We report both k-NN and linear evaluation for CPC [110], SimCLR [31], MoCo [64, 32, 34], BT [157], SwAV [22], BYOL [59], BEiT [12], DINO [23, 111], and iBOT [163]. (a) Timeline of the overall best performance on linear evaluation. (b) k-NN and (c) linear evaluation for various architectures with the number of parameters.

An overview of the top performing models in the SSL field is depicted in Figure 2.10. We highlight the evolution of the models' performances on ImageNet-1k, which is the reference dataset for SSL evaluation. The dataset includes a large variety of $1,000$ classes, such as vehicles, dog breeds, vegetables, or even furniture. The images do not depict medical tissue. However, the span and variety of classes are large enough to give a good approximation of the model's global performances. The top view depicts the evolution of the best top-1 linear evaluation accuracies across the years. The bottom plots give a more detailed performance overview as a function of the work, encoder, and number of parameters used for k-NN and linear evaluations. Linear evaluation is defined as a single linear layer trained on top of the pre-trained encoder, while k-NN is defined as the direct evaluation of the embedding space (raw features).

In subsection 2.4.1, subsection 2.4.2, and subsection 2.4.3, we introduce the most common SSL approaches as constrastive learning (CL), correlation/clustering-based, and self-distilation, respectively. In subsection 2.4.4, we present the concept of self-supervision



(a) Anchor ($\mathbf{z}$) (b) Positive ($\mathbf{z}^+$) (c) Negative ($\mathbf{z}^-$) (d) Contrastive learning

Figure 2.11 – Illustration of constrastive learning applied to histological patches. (a) The reference image is named anchor. (b-c) The anchor forms a positive (visually similar) and negative pair (visually dissimilar). (d) The optimization process tries to maximize the similarity of the anchor to the positive example while repelling the negative one.

using auxiliary tasks and its application to histological images. Finally, we present the latest trends in SSL for histological data subsection 2.4.5. For an extended introduction to SSL, we recommend the works of [10, 36].

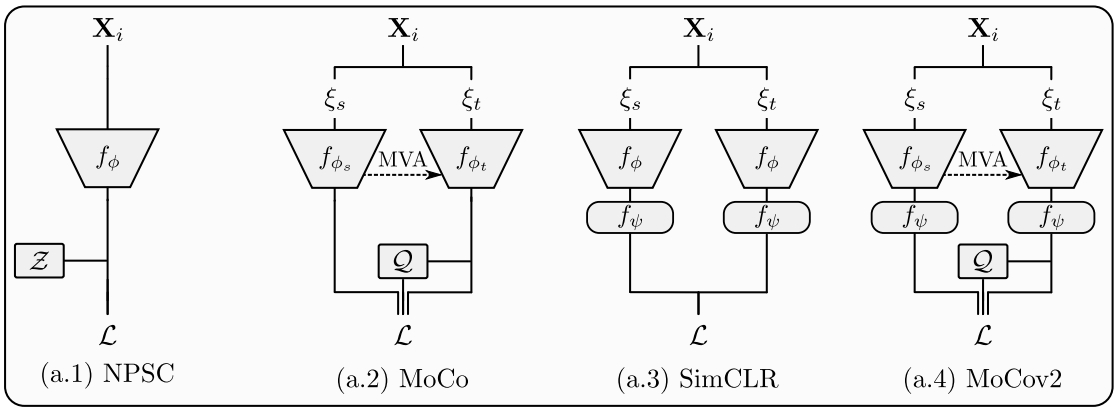
2.4.1 Contrastive Learning

A wide variety of SSL models rely on the principle of CL to learn feature embedding. We start with an image, called an anchor, representing any data (*e.g.* image of a tumor). Then, two additional images are selected. The first one is called the positive image and is assumed to share visual similarity with the anchor (*e.g.* another example of tumor). The second is called negative and is, on the contrary, visually different from the original anchor (*e.g.* a muscle tissue). CL aims at increasing the feature similarity between the anchor and positive example while maintaining disparity with the negative sample, as depicted in Figure 2.11.

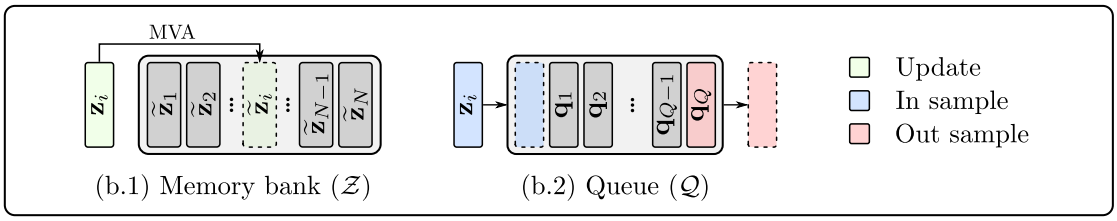
More formally, we assume we have access to multiple RGB images $\mathbf{X}_i \in \mathbb{R}^{H\times W\times 3}$ from a dataset $\mathcal{W}$ such that $\{\mathbf{X}_i\}_{i=1}^N = \mathcal{W}$, where H and W denote the width and height of the image, N the size of the dataset, and i the index of the image within the dataset. For each image, we extract its embedding using an encoder $f_\phi : \mathbb{R}^{H\times W\times 3} \to \mathbb{R}^D$ as $\mathbf{z}_i = f_\phi(\mathbf{X}_i)$, where D is the size of the embedding, and ϕ the model's learnable parameters. In addition to the images and their embedding, we assume we have access to two sets of labeled pair indexes that indicate whether two samples are considered positives (similar) or negatives (dissimilar):

$$\mathcal{I} = \{(i,j) \mid \mathbf{X}_i, \mathbf{X}_j \text{ are similar}\}, \quad \text{and} \quad \backslash\mathcal{I} = \{(i,j) \mid \mathbf{X}_i, \mathbf{X}_j \text{ are dissimilar}\}. \tag{2.20}$$

One of the first significant works on CL uses it for face verification [26, 35]. The



(a) Contrastive learning architectures



(b) Maintaining negative samples

Figure 2.12 – Architectures using contrastive learning to learn feature representation. (a) Architectures NPSC [151], MoCo [64], SimCLR [31], and MoCov2 [32] for loss optimization $\mathcal{L}$ given a input image sample $\mathbf{X}_i$, augmentations ξ_s, ξ_t, encoder (student, teacher) f_ϕ (f_{ϕ_s}, f_{ϕ_t}), projection head f_ψ, queue $\mathcal{Q}$, memory bank $\mathcal{Z}$, and moving average (MVA). (b) Different ways to maintain and update the set of negative samples for CL given a new embedding $\mathbf{z}_i$.

optimization term is built such that it tries to reduce the distance between the anchor (reference face) and the positive sample (face from the same person) while maximizing the distance to the negative one (face of an impostor). The loss is named triplet loss and relies on three components (*i.e.* anchor, positive, and negative samples). It is defined as:

$$\min \mathcal{L}_{\text{triplet}} = \min_{\phi} \sum_{(i,j)\in\mathcal{I}} \underbrace{\|\mathbf{z}_i - \mathbf{z}_j\|_2}_{\text{Positive pair}} + \sum_{(i,j)\in\setminus\mathcal{I}} \max(0, \underbrace{\alpha}_{\text{Margin}} - \underbrace{\|\mathbf{z}_i - \mathbf{z}_j\|_2}_{\text{Negative pair}}), \tag{2.21}$$

where $\alpha \in \mathbb{R}_+$ is a tolerance margin acting as a threshold to avoid the collapsing of the representation (*i.e.* convergence to constant vectors). The Euclidean distance measures the similarity of positive and negative pairs. To minimize the loss, the model needs to reduce the positive pair distance while maximizing the negative ones.

As the research progresses, the triplet loss is iteratively updated. The Euclidean distance and positive part function are progressively replaced by the cosine similarity and the exponential function [62, 110, 151]. In Figure 2.12, we highlight the evolution of the

recent methods for CL and their way of handling negative entries. In NPSC [151], they use a simple architecture $\mathbf{z}_i = f_\phi(\mathbf{X}_i)$ and a noise-contrastive estimation (NCE) loss to learn feature representations:

$$P(\mathbf{z}, \mathbf{z}^+, \mathbf{Z}^-) = \frac{\overbrace{\exp \frac{\mathbf{z}^\top \mathbf{z}^+ / \tau}{\|\mathbf{z}\|_2 \|\mathbf{z}^+\|_2}}^{\text{Positive pair}}}{\underbrace{\exp \frac{\mathbf{z}^\top \mathbf{z}^+ / \tau}{\|\mathbf{z}\|_2 \|\mathbf{z}^+\|_2}}_{\text{Positive pair}} + \sum_{\mathbf{z}^- \in \mathbf{Z}^-} \underbrace{\exp \frac{\mathbf{z}^\top \mathbf{z}^- / \tau}{\|\mathbf{z}\|_2 \|\mathbf{z}^-\|_2}}_{\text{All negative pairs}}}, \tag{2.22}$$

$$\min \mathcal{L}_{\text{NCE}} = \min_\phi \sum_{i=1}^{N} - \left[\log(P(\mathbf{z}_i, \widetilde{\mathbf{z}}_i, \mathcal{Z} \backslash \widetilde{\mathbf{z}}_i)) + \sum_{k=1, k \neq i}^{N} \log(1 - P(\widetilde{\mathbf{z}}_i, \widetilde{\mathbf{z}}_k, \mathcal{Z} \backslash \widetilde{\mathbf{z}}_i)) \right], \tag{2.23}$$

where $\tau \in \mathbb{R}_+$ is called the temperature and controls the sharpness of the confidence predictions [145] and $\mathcal{Z} = \{\widetilde{\mathbf{z}}_i \in \mathbb{R}^D\}_{i=1}^N$ the memory bank that keeps track of all samples embedding. When a new sample $\mathbf{z}_i$ is produced by the encoder, its respective entry in the memory bank is updated using moving average (MVA):

$$\widetilde{\mathbf{z}}_i \leftarrow (m)\widetilde{\mathbf{z}}_i + (1 - m)\mathbf{z}_i, \tag{2.24}$$

where $m \in]0, 1]$ is the momentum and fixes the importance given to new samples. The memory bank is used to sample negative entries for the NCE loss. The model tries to maximize the similarity of the i-th sample $\mathbf{z}_i$ and its memory embedding $\widetilde{\mathbf{z}}_i$ while minimizing its similarity to all other memory bank entries $\mathcal{Z} \backslash \widetilde{\mathbf{z}}_i$. In this setting, the size of the memory bank $|\mathcal{Z}| = N$ is tied to the number of samples in the dataset. When working with histological data, datasets can include millions of examples, which become hard to maintain memory-wise.

In *Momentum Contrast* (MoCo) [32], a new milestone is reached with the reformulation of the problem. The model uses random transformations to generate positive entries. Given an input image $\mathbf{X}_i$, two sets of random transformations $\xi_s, \xi_t : \mathbb{R}^{H \times W \times 3} \to \mathbb{R}^{H \times W \times 3}$ and two encoders $f_{\phi_s}, f_{\phi_t} : \mathbb{R}^{H \times W \times 3} \to \mathbb{R}^D$ with parameters ϕ_s, ϕ_t, we can extract a positive pair embedding as $(\mathbf{z}_i, \mathbf{z}_i^+)$, where $\mathbf{z}_i = f_{\phi_s}(\xi_s(\mathbf{X}_i))$ and $\mathbf{z}_i^+ = f_{\phi_t}(\xi_t(\mathbf{X}_i))$. Since both terms come from an augmented version of the same image, they are, by definition, similar. For the negative sample, a different strategy is considered. Instead of relying on a memory bank, MoCo maintain a queue $\mathcal{Q} = \{\mathbf{q}_k \in \mathbb{R}^D\}_{k=1}^Q$ of negative samples. The queue length is fixed and independent of the dataset size with $Q \ll N$. The model maximizes the similarity between the reference view and its augmentation while considering all other

samples as dissimilar. The loss becomes:

$$\min \mathcal{L}_{\text{infoNCE}} = \min_{\phi_s} \sum_{i=1}^{N} P(\mathbf{z}_i, \mathbf{z}_i^+, \mathcal{Q}). \tag{2.25}$$

The queue is maintained using a first in, first out logic. Moreover, the weight ϕ_s are optimized using backpropagation while the weight of ϕ_t are updated using MVA:

$$\begin{cases} \mathbf{q}_1 \leftarrow \mathbf{z}^+ \\ \mathbf{q}_k \leftarrow \mathbf{q}_{k-1} \quad , \forall k \neq 1, k < Q \end{cases}, \tag{2.26}$$

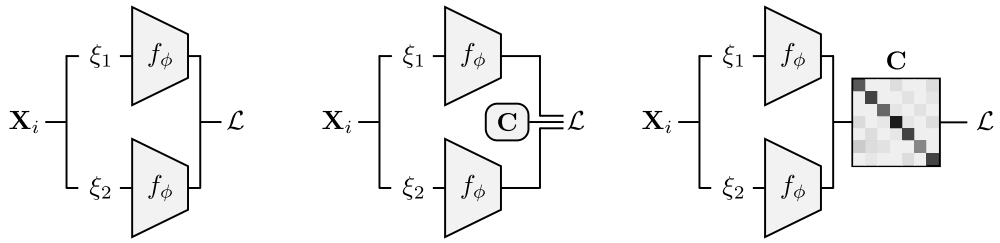
$$\phi_t \leftarrow (m)\phi_t + (1-m)\phi_s \qquad m \in [0,1]. \tag{2.27}$$

The work of MoCo is followed by multiple works exploring other aspects of CL. In *a Simple framework for Contrastive Learning of visual Representations* (SimCLR) [31], a projection head $f_\psi : \mathbb{R}^D \rightarrow \mathbb{R}^{D'}$ with parameters ψ is added on top of the encoder to improve the feature representation. The output dimension D' is typically kept lower than the output encoder dimension $D' \leq D$. Moreover, the architecture uses a single encoder f_ϕ to embed features. The work of SimCLR does not rely on a memory bank to keep track of the negative examples. Here, the model considers the elements of the batch as negative entries. As a result, the model requires large batches to be trained, which is a major limitation. In addition, the work highlights the critical importance of selecting data augmentation operators and, in particular, using random cropping with resizing, Gaussian blurring, and color jittering.

Later, MoCov2 [32] adapts its architecture to match SimCLR findings by including a projection head. Finally, the latest MoCov3 uses ViTs to improve feature descriptors further and remove the use of the queue. By the time of the writing, all follow-up works on CL mainly focus on small process optimization, while the core concept remains unchanged.

2.4.2 Correlation and Clustering

In this section, we describe the use of two peculiar SSL methods. They are inspired by CL but differ from it. They use a clustering approach in *Swapping Assignments between multiple Views* (SwAV) [22] and a correlation function in Barlow Twins (BT) [157] to learn feature representation. In Figure 2.13, we show the difference in architecture between the standard CL and the presented models. Both works rely on two sets of transformations $\xi_1, \xi_2 : \mathbb{R}^{H \times W \times 3} \rightarrow \mathbb{R}^{H \times W \times 3}$ and an encoder $f_\phi : \mathbb{R}^{H \times W \times 3} \rightarrow \mathbb{R}^D$ with



(a) Contrastive learning (b) SwAV (c) Barlow Twins (BT)

Figure 2.13 – Comparison between standard CL architecture, SwAV and BT given input image $\mathbf{X}_i$, augmentations ξ_1, ξ_2, encoder f_ϕ, and model-dependent matrix $\mathbf{C}$. SwAV use a cluster-based approach while BT use a correlation-based one. The definition of the loss $\mathcal{L}$ is model dependent.

parameters ϕ.

We start with SwAV and its clustering procedure. Instead of relying on negative samples, they try to cluster the embeddings into a limited number of clusters. To do so, they define a learnable clustering matrix $\mathbf{C} \in \mathbb{R}^{D \times K}$, where D is the size of the embedding space and K is the number of clusters. The model uses the matrix $\mathbf{C}$ to project the embeddings to a lower dimensional space. The result of the projection is defined as the output probability map $\mathbf{P} = (p_{i,k})_{1 \leq i \leq N, 1 \leq k \leq K}$:

$$p_{i,k} = \frac{\exp(\mathbf{z}_i^\top \mathbf{c}_k / \tau)}{\sum_{k'=1}^{K} \mathbf{z}_i^\top \mathbf{c}_{k'} / \tau}, \tag{2.28}$$

where τ is the temperature factor, $p_{i,k}$ is the probability that the i-th sample belongs to cluster k, and both the embedding and cluster centers are normalized (*i.e.* $\|\mathbf{z}_i\|_2 = \|\mathbf{c}_k\|_2 = 1$). Moreover, the outputs are passed through a softmax function to ensure a sharp prediction. During parameters optimization, the model tries to align the probability assignment $\mathbf{P}$ to a target distribution $\mathbf{Q} = (q_{i,k})_{1 \leq i \leq N, 1 \leq k \leq K}$ via the loss:

$$\min \mathcal{L}_{\text{SWAV}} = \min_{\phi} - \sum_{i=1}^{N} \sum_{k=1}^{K} q_{i,k} \log(p_{i,k}). \tag{2.29}$$

The target distribution is based on the positive entries $\mathbf{Z}^+ = (\mathbf{z}_1^+, \cdots, \mathbf{z}_N^+)$ which represent the same image embedding but augmented through a different set of transformations. The target probability is estimated at every optimization step as:

$$\max_{\mathbf{Q} \in \mathcal{Q}} = \text{Tr}(\mathbf{Q}\mathbf{C}^\top \mathbf{Z}^+) + \epsilon H(\mathbf{Q}), \tag{2.30}$$

where H is the entropy function, ϵ the parameter that controls the smoothness (*e.g.* $\epsilon = 0.05$), Tr the trace of the matrix, and $\mathcal{Q}$ the constrains on the target distribution. The optimization of Equation 2.30 is based on the Sinkhorn-Knopp [38] algorithm. For more information, please refer to SwAV original work [22]. The goal of the constraints on the target probabilities is to reduce the risk of cluster collapse by enforcing a uniform distribution of the embedding into the K clusters.

Aside from its cluster-based approach, one of the main contributions of SwAV is the introduction of the concept of multi-crop views that greatly help the learning of feature similarity between positive pairs. The authors mention that random cropping is critical to learning localized information between views. However, increasing the number of views also increases the memory requirements. To tackle this issue, they propose to use a standard image (*e.g.* 224px $\times$ 224px) as a reference and to sample a few additional low-resolution crops from it (*e.g.* 96px $\times$ 96px). The model then compares the embedding of the cropped areas to the original image. As the allocated memory quadratically increases with the image size, using low-resolution crops is negligible.

The second method presented in this section is the work of BT [157]. The authors replace the use of negative samples with the computation of feature cross-correlation as $\mathbf{C} = (c_{j,j'})_{1\leq j,j'\leq D}$. Let's define the output predictions $\mathbf{Z} = (z_{i,j})_{1\leq i\leq N, 1\leq j\leq D} = (\mathbf{z}_1, \cdots, \mathbf{z}_N)^\top$ and $\mathbf{Z}^+ = (z^+_{i,j})_{1\leq i\leq N, 1\leq j\leq D} = (\mathbf{z}^+_1, \cdots, \mathbf{z}^+_N)^\top$. The correlation between two features j and j' is given as:

$$c_{j,j'} = \frac{\sum_{i=1}^{N}(z_{i,j} - \mu_j)(z^+_{i,j'} - \mu^+_{j'})}{\sqrt{\sum_{i=1}^{N}(z_{i,j} - \mu_j)^2}\sqrt{\sum_{i=1}^{N}(z^+_{i,j'} - \mu^+_{j'})^2}}, \qquad j, j' \in \{1, \ldots, D\}, \tag{2.31}$$

$$\mu_j = \frac{1}{N}\sum_{i=1}^{N} z_{i,j}, \qquad \text{and} \qquad \mu^+_j = \frac{1}{N}\sum_{i=1}^{N} z^+_{i,j}.$$

Let's now assume that the model achieves an optimal feature representation. The influence of the data augmentation, therefore, should be negligible. Specifically, the correlation matrix between the two sets should be diagonal to maximize inner similarity while minimizing cross-correlation between features. The BT loss is then defined as:

$$\min \mathcal{L}_{\text{BT}} = \min_{\phi} \sum_{j=1}^{D} \underbrace{(1 - c_{j,j})^2}_{\text{inner term}} + \sum_{j=1}^{D}\sum_{j'\neq j} \underbrace{{c_{j,j'}}^2}_{\text{cross term}} .$$

What makes BT an interesting approach is that instead of considering the correlation

along the feature dimension, they investigate the correlation along the batch dimension. As a result, the model performs better than SimCLR when confronted with small batch sizes. Moreover, minimizing the cross-term encourages feature decorrelation and reduces information redundancy. We can see the method as trying to maximize positive sample correlation (diagonal terms) while repelling negative ones (non-diagonal terms).

2.4.3 Self-Distillation

Recently, the development of novel methods using CL slowed down. New architectures only significantly improve feature representations by increasing models' computational complexities [118]. The renewal of SSL comes with the emergence of (self-)distillation architectures. Distillation models are composed of two branches working together. The first branch, called the student model, is typically small and compact to synthesize information. The second branch, called the teacher model, tends to be more complex. The fundamental intuition behind knowledge distillation is that large models fail to exploit their capacity fully. As a result, we try to transfer (*i.e.* distillate) the information from the large model (*i.e.* teacher) to the small one (*i.e.* student) while maintaining competitive performances. When the teacher and student branches are identical, we talk about self-distillation [5].

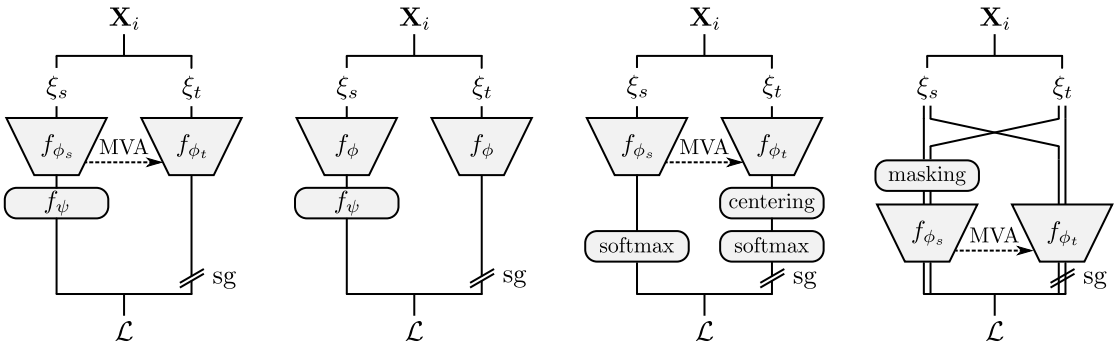
We present the most common self-distillation models in chronological order. To learn feature representation, the models use different metrics to maximize information between two views. They differ from CL approaches as they do not rely on negative example support. The presented architectures are listed in Figure 2.14 along with their core concept. We define as f_{ϕ_s}, $f_{\phi_t} : \mathbb{R}^{H\times W\times 3} \to \mathbb{R}^D$ the student and teacher networks with parameters ϕ_s and ϕ_t respectively. The augmentation functions $\xi_s, \xi_t : \mathbb{R}^{H\times W\times 3} \to \mathbb{R}^{H\times W\times 3}$ are used to create variation of an input image $\mathbf{X}_i$.

One of the first breakthroughs of self-distillation models comes with *Bootstrap Your Own Latent* (BYOL) [59]. The logic is identical to SimCLR and MoCo, where the augmentation of the view serves as positive pair examples. The optimization loss in BYOL is:

$$\min \mathcal{L}_{\text{BYOL}} = \min_{\phi_s,\psi} -\sum_{i=1}^{N} \frac{f_\psi(\mathbf{z}_i^s)^\top \mathbf{z}_i^t}{\|f_\psi(\mathbf{z}_i^s)\|_2 \|\mathbf{z}_i^t\|_2}, \tag{2.32}$$

$$\mathbf{z}_i^s = f_{\phi_s}(\xi_s(\mathbf{X}_i)), \qquad \text{and} \qquad \mathbf{z}_i^t = f_{\phi_t}(\xi_t(\mathbf{X}_i)),$$

where $f_\psi : \mathbb{R}^D \to \mathbb{R}^D$ is a multilayer predictor with parameters ψ. The predictor is applied to the student output predictions to create an asymmetry between the branches and avoid collapsing (*i.e.* model predicting the same output on both sides). Using



(a) BYOL (b) SimSiam (c) DINO (d) iBOT

Figure 2.14 – Latest self-distillation architectures. (a) BYOL [59], (b) SimSiam [33], (c) DINO(v2) [23, 111], and (d) iBOT [163]. We denote the input data as $\mathbf{X}_i$, data augmentations as ξ_s, ξ_t, encoder (student, teacher) as f_ϕ (f_{ϕ_s}, f_{ϕ_t}), projection head as f_ψ, stop-gradient as sg, and moving average as MVA. The definition of the loss $\mathcal{L}$ is model-dependent.

standard backpropagation, we first update the student branch (ϕ_s, ψ). Then, the weights of the teacher model (ϕ_t) are updated using a MVA:

$$\phi_t \leftarrow m\phi_t + (1-m)\phi_s \qquad m \in [0,1]. \tag{2.33}$$

A surprising aspect of BYOL is its resilience to collapse. Mathematically, the model admits trivial solutions (*e.g.* collapsing of the predictions to constant vector). However, the authors empirically prove that their approach never converges to such a solution. As a result, the authors hypothesize that the combination of the predictor and MVA prevent the model from collapsing. In SimSiam [33], they further investigate the stability of BYOL through various experiments. They prove that the MVA does not improve and even deteriorates the embedding quality. As a result, the MVA is replaced with a simple weight sharing (*i.e.* $\phi_s = \phi_t = \phi$).

The work of SimSiam and BYOL is followed by *knowledge DIstillation with NO labels* (DINO) [23]. Instead of cosine similarity, they rely on cross-entropy to learn feature embedding. As the previous model, DINO uses various tricks to avoid collapsing. They propose the use of both centering and output sharpening to regularize convergence. The loss is given as:

$$\min \mathcal{L}_{\text{DINO}} = \min_{\phi_s} -\sum_{i=1}^{N} H(\text{softmax}(\mathbf{z}_i^s/\tau_s), \text{softmax}((\mathbf{z}_i^t - \mathbf{c})/\tau_t)), \tag{2.34}$$

where $\tau_s, \tau_t \in \mathbb{R}_+$ are temperature factors, H is the cross-entropy function, and $\mathbf{c}$ the

output centers. The temperature are typically small (*e.g.* $\tau_s, \tau_t \leq 0.1$). The centers are learnable parameters and are used to "prevent one dimension from dominating" over the other ones. Both the parameters of the teacher model and centers are updated at each training step using MVA. Moreover, DINO is the first major self-distillation work to take advantage of ViT encoders to boost its performance. ViT-based models show better accuracy than previous ResNet-based encoders.

Later, *image BERT pre-training with Online Tokenizer* (iBOT) [163] proposes a novel composition of two loss terms to learn feature embedding. The first term $\mathcal{L}_{\text{CLS}}$ is based on a cross-view while the second term $\mathcal{L}_{\text{MIM}}$ is based on an in-view optimization. Moreover, the approach relies on architectural specificities of ViTs. Before an image is fed to the ViT encoder, it is split into tiny image crops. For example, according to the standard terminology, a model named ViT/16 means the input image is cut into small 16×16 pieces. In addition to these crops, ViTs append a learnable class token to the list of inputs. As a result, both the crops and the class token are processed by the model. In practice, we use the output prediction of the class token as a reference for the image embedding rather than the individual crop outputs. For more information about the behavior of ViTs, we recommend the work of [47].

More formally, let's assume we have an input image $\mathbf{X}_i \in \mathbb{R}^{H \times W \times 3}$. We split the input image into small patches $\mathbf{P}_{i,j} \in \mathbb{R}^{H/P \times W/P \times P^2}$, where P^2, H/P, W/P are the number, height, and width of the patches and $j \in \{1, \cdots, P^2\}$ is the patch index. Therefore, $\mathbf{P}_{i,j}$ denotes the j-th patch of the i-th image. Subsequently, we define as $\mathbf{z}_i \in \mathbb{R}^D$ the class embedding of an image $\mathbf{X}_i$ and as $\mathbf{v}_{i,j} \in \mathbb{R}^D$ the embedding of the patch $\mathbf{P}_{i,j}$. The outputs of the student model are given as the image and patches embedding using ξ_s, ξ_t data augmentations:

$$\begin{aligned} \mathbf{z}_i^s = f_{\phi_s}(\text{masking}(\xi_s(\mathbf{X}_i))), \quad \hat{\mathbf{z}}_i^s = f_{\phi_s}(\text{masking}(\xi_t(\mathbf{X}_i))), \\ \mathbf{v}_{i,j}^s = f_{\phi_s}(\text{masking}(\xi_s(\mathbf{P}_{i,j}))), \quad \text{and} \quad \hat{\mathbf{v}}_{i,j}^s = f_{\phi_s}(\text{masking}(\xi_t(\mathbf{P}_{i,j}))), \end{aligned} \tag{2.35}$$

where the masking operator randomly replaces a subset of the image patches with mask tokens. The output is, therefore, a corrupted version of the original input. The logic is similar to the one of the dropout [68] where part of the information is removed to increase the model robustness. For the teacher branch, the output definitions are the same, except that no masking is applied:

$$\begin{aligned} \mathbf{z}_i^t = f_{\phi_t}(\xi_s(\mathbf{X}_i)), \quad \hat{\mathbf{z}}_i^t = f_{\phi_t}(\xi_t(\mathbf{X}_i)), \\ \mathbf{v}_{i,j}^t = f_{\phi_t}(\xi_s(\mathbf{P}_{i,j})), \quad \text{and} \quad \hat{\mathbf{v}}_{i,j}^t = f_{\phi_t}(\xi_t(\mathbf{P}_{i,j})). \end{aligned} \tag{2.36}$$

The overall loss mixes the prediction from both branches and augmentations. The iBOT loss is then given as the summation of the cross and in-view losses:

$$\min \mathcal{L}_{\text{iBot}} = \min_{\phi_s} \overbrace{-\sum_{i=1}^{N} (\mathbf{z}_i^t)^\top \log(\hat{\mathbf{z}}_i^s) + (\hat{\mathbf{z}}_i^t)^\top \log(\mathbf{z}_i^s)}^{\text{Cross-view loss } (\mathcal{L}_{\text{CLS}})} \underbrace{- \sum_{i=1}^{N} \sum_{j=1}^{P^2} m_{i,j} \, (\mathbf{v}_{i,j}^t)^\top \log(\mathbf{v}_{i,j}^s) + m_{i,j} \, (\hat{\mathbf{v}}_{i,j}^t)^\top \log(\hat{\mathbf{v}}_{i,j}^s)}_{\text{In-view loss } (\mathcal{L}_{\text{MIM}})}, \quad (2.37)$$

where $m_{i,j} \in 0, 1$ is the masking parameter that keeps track of the patch masked at the input.

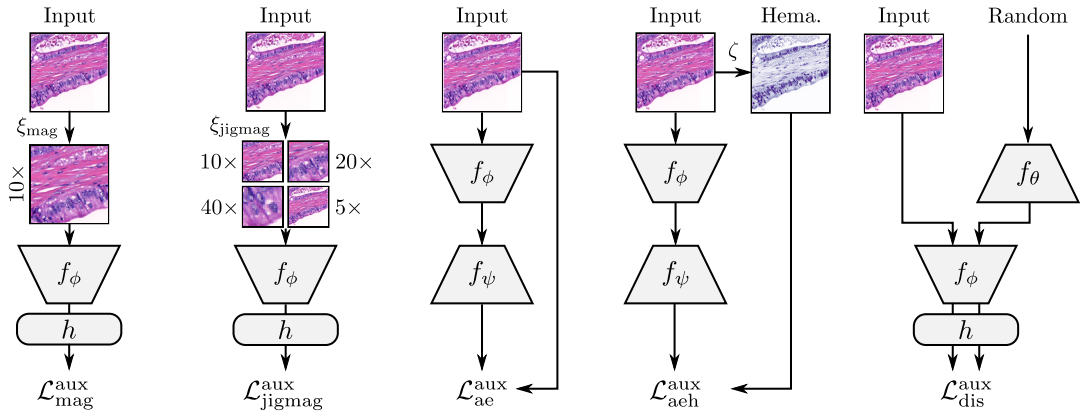
At the time of writing, the state of the art (SOTA) model in SSL is DINOv2 [111]. Fundamentally, the DINOv2 architecture does not include any significant contribution with respect to DINO. Still, the model cleverly takes advantage of recent works to boost its performance. Out of all the minor modifications proposed, we identify the three main components as (i) the implementation of the recent improvements proposed in iBOT, (ii) the enlargement of batch sizes, and (iii) the use of curated data (*i.e.* sets of data that are filtered to match specific queries and tasks).

2.4.4 Auxiliary Tasks

In this section, we introduce the concept of auxiliary tasks that are external cost functions. Such tasks are typically self-supervised and often data-specific. Examples of auxiliary tasks include the prediction of image rotation [56], channels conversion to different color spaces [158], patches localization within the image [46], image reconstruction through autoencoders [143], inpainting [113], or jigsaw puzzle solving [108]. Auxiliary tasks $\mathcal{L}_i^{\text{aux}}$ are added on top of an already existing primary loss $\mathcal{L}_{\text{primary}}$ term (*e.g.* classification or segmentation) to form a constrained loss $\mathcal{L}$:

$$\mathcal{L} = \mathcal{L}_{\text{primary}} + \underbrace{\sum_i \mathcal{L}_i^{\text{aux}}}_{\text{Auxiliary tasks}} . \quad (2.38)$$

Hence, removing auxiliary terms does not hinder the convergence capability of the main term. Their core purpose is to help the primary loss term to converge toward better feature representation. It is a particular case of multi-task learning. The subtlety is that



(a) Magnification (b) JigMag (c) Autoencoder (d) Hematoxylin (e) Generative

Figure 2.15 – Examples of auxiliary tasks for SSL [85, 130]. (a) Simple prediction of magnification. (b) Jigsaw puzzle with various magnifications. (c-d) Autoencoder architecture to reconstruct RGB or hematoxylin channels. (e) Generative model and discriminator. We denote the encoder, decoder, generator, classifier/discriminator, data augmentation, and stain extraction as f_ϕ, f_ψ, f_θ, h, ξ, and ζ respectively.

multi-task learning does not prioritize one loss over the other, whereas here, we consider one loss as the primary and all others as optional.

As mentioned before, the design of auxiliary tasks is often tied to the data structure. For example, let's consider the picture of a cat lying on the ground. Here, the concept of up and down is directly defined within the image. Hence, rotating the image and asking the network to retrieve the original angle would make sense. However, when considering a histological picture, the concept of up and down is not defined as the data lack orientation. In this setting, using rotation as an additional task would not improve the feature representation.

Recent works [85, 130] analyze the relevance of various auxiliary tasks applied to histological data. The tasks that show the best performances are presented in Figure 2.15. The first task is the prediction of magnification. Given an input image $\mathbf{X}_i \in \mathbb{R}^{H\times W\times 3}$ we aim to predict whether the image is acquired at, for example, $5\times, 10\times, 20\times$, or $40\times$. To do so, we use a simple encoder $f_\phi : \mathbb{R}^{H\times W\times 3} \to \mathbb{R}^D$ and classifier $h : \mathbb{R}^D \to \mathbb{R}^C$, where D is the size of the embedding space and C the number of possible magnifications. The loss is given as:

$$\min \mathcal{L}^{\text{aux}}_{\text{mag}} = \min_{f_\phi, h} \sum_{i=1}^{N} H(h \circ f_\phi \circ \xi_{\text{mag}}(\mathbf{X}_i), \mathbf{y}_i), \tag{2.39}$$

where $\mathbf{y}_i \in \mathbb{R}^C$ is the coded magnification label, H the cross entropy function, N the total number of samples, and $\xi_{\text{mag}} : \mathbb{R}^{H\times W\times 3} \to \mathbb{R}^{H\times W\times 3}$ the magnification transformation.

In practice, WSIs include different magnifications available for the same image. However, if the target magnification is not available, the transformation is replaced by a simple image interpolation.

Another possible task is jigsaw puzzle solving. However, histological images lack ordering as well. For example, different tumor types have distinct tumor/stroma arrangements. To solve this, the authors propose replacing the jigsaw with a mosaic of the image at different magnifications. The model tries to predict which magnification is present in each part of the image. The loss is given as:

$$\min \mathcal{L}^{\text{aux}}_{\text{jigmag}} = \min_{f_\phi, h} \sum_{i=1}^{N} H(h \circ f_\phi \circ \xi_{\text{jigmag}}(\mathbf{X}_i), \mathbf{y}_i), \tag{2.40}$$

and is highly similar to the magnification prediction. The differences lie in the augmentation $\xi_{\text{jigmag}} : \mathbb{R}^{H \times W \times 3} \to \mathbb{R}^{H \times W \times 3}$ that generates the mosaic and the number of output classes C for the label. Given the four possible magnifications, we end up with $C = 4! = 24$ combinations.

The third and fourth tasks explored are the use of an autoencoder [67]. Here, the encoder is coupled with a decoder $f_\psi : \mathbb{R}^D \to \mathbb{R}^{H \times W \times 3}$ that projects the encoded image back to the input space. The loss is then given as the difference between the reconstructed image and the input one:

$$\min \mathcal{L}^{\text{aux}}_{\text{ae}} = \min_{f_\phi, f_\psi} \sum_{i=1}^{N} \| f_\psi \circ f_\phi(\mathbf{X}_i) - \mathbf{X}_i \|_2. \tag{2.41}$$

One of the main drawbacks of using autoencoders is that we have no control over the quality of the encoded results. When training an autoencoder, we want the first stage (encoder) to capture most of the information, as the decoder will be dropped for downstream tasks. We usually use shallow decoders (*e.g.* fewer parameters in decoder than in encoder) to ensure a strong feature representation through the encoder. Due to its higher complexity, the encoder will balance the sloppiness of the decoder.

The other alternative use of the encoder is to focus on a single channel. Instead of reconstructing the RGB image, we target one of the HE channels. The reason is that the RGB to RGB reconstruction does not ensure the learning of the structural aspect of the image. The autoencoder might only learn interpolation functions to reconstruct images. By converting the input to HE channels, we force the model to capture the picture's underlying structure [4]. Out of the HE channels, hematoxylin captures most

of the information with cell nuclei [85]. The loss is then given as:

$$\min \mathcal{L}_{\text{aeh}}^{\text{aux}} = \min_{f_\phi, f_\psi} \sum_{i=1}^{N} \| f_\psi \circ f_\phi(\mathbf{X}_i) - \zeta(\mathbf{X}_i) \|_2, \tag{2.42}$$

where $\zeta : \mathbb{R}^{H \times W \times 3} \rightarrow \mathbb{R}^{H \times W}$ is the function that converts the RGB image to its hematoxylin equivalent. The decoder output is adjusted to fit the number of channels of the reconstructed hematoxylin image.

The last approach presented is the use of a generative network [37]. Here, an additional branch tries to fool the network by generating fake inputs. The network $f_\theta : \mathbb{R}^Z \rightarrow \mathbb{R}^{H \times W \times 3}$ takes as input a random variable $\mathbf{z} \in \mathbb{R}^Z$ to generate a fake image, where Z is the size of the random vector. The fake images are fed with the real data to fool the classifier h. The discriminator loss is given as:

$$\min \max \mathcal{L}_{\text{dis}}^{\text{aux}} = \min_{f_\theta} \max_{f_\phi, h} \sum_{i=1}^{N} \underbrace{\log(h \circ f_\phi(\mathbf{X}_i))}_{\text{Real}} + \sum_{j=1}^{M} \underbrace{\log(1 - h \circ f_\phi \circ f_\theta(\mathbf{z}_j))}_{\text{Fake (generated)}}, \tag{2.43}$$

where we train the encoder and discriminator to maximize the probability of correct prediction while optimizing the generator to fool the discriminator.

All the presented tasks are optional and can be added to an existing architecture to boost its performance. Moreover, they are all self-supervised tasks and do not require additional annotations. It comes particularly handy when only a subset of the data is labeled. The training of models using auxiliary tasks shows competitive results with fully supervised methods [85]. More specifically, the use of magnification, jigmag, and generative models largely contribute to the performance boost.

The auxiliary tasks mentioned in this section are a non-exhaustive list of possible tasks applied to histological data. Other approaches take advantage of the inner properties of histological images. We cite, for example, the use of spatial proximity in WSIs to build positive training pairs [4, 137], the use of different stain augmentation to make the model invariant to stain variations [135], or taking advantage of the rotation-agnostic aspect of histological images to train rotation equivariant networks [142].

2.4.5 Current Research

Finally, we conclude this section by reviewing the current SOTA for SSL in histology. Overall, medical research tends to follow the computer vision field. It is common to see

Table 2.2 – Results of linear evaluation and fine-tuning over multiple histological datasets (BACH [8], K19 [77], PCAM [142], and MHIST [148]) as taken from [75]. The fraction of labels used for the evaluation is reported for each experiment.

		Linear evaluation				Fine-tuning		
		BACH	K19	PCAM	MHIST	K19		
Arch.	Method	100%	100%	100%	100%	1%	10%	100%
ResNet-50	Supervised	80.83	90.93	80.79	76.25	90.28	93.87	92.09
	MoCov2 [32]	77.50	93.52	86.78	77.07	**91.73**	**95.10**	**96.21**
	SwAV [22]	83.33	**95.78**	85.28	71.14	89.26	92.84	93.31
	BT [157]	**87.50**	94.60	**88.15**	**78.81**	91.23	92.84	93.23
ViT-S/16	Supervised	75.83	91.56	80.96	**78.51**	93.15	94.76	95.81
	DINO [23]	**85.83**	94.19	88.78	76.15	94.03	94.92	95.81
ViT-S/8	DINO [23]	83.33	**95.29**	**90.12**	77.89	**95.03**	**96.27**	**97.13**

the emergence of a novel SSL work followed by its application to medical images in the next months. We cite for example the use of SSL-based methods using autoencoders [4, 155], MoCo [1, 147], SimCLR [88], BYOL [146], DINO [30], or iBOT [51] applied to histopathology.

When applying SOTA SSL models to the medical fields, authors tend to add small tricks on top of the base method. Although those modifications aim to boost their approaches' performance, it becomes difficult to quantify the contribution of each component with respect to the original SSL work. A recent benchmark [75] compares the raw performances of the recent SOTA SSL models over different histological datasets.

The first important outcome of their research is that self-supervised learning outperforms the standard supervised approach. In other words, a model pre-trained on medical data achieves better performances on the same medical data compared to their ImageNet pre-trained counterparts. This conclusion is more complex than it seems. In many medical applications, such as MRI, images are expensive to acquire and not abundant. For histology, HE slides are relatively cheap to process and available in large quantities. It is only because of their abundance that we can reliably train self-supervised models for downstream tasks.

Secondly, they present the synthesized results as presented in Table 2.2 on multiple histological datasets (BACH [8], K19 [77], PCAM [142], and MHIST [148]). For the supervised reference, the authors considered pre-trained weights on ImageNet. For DINO, they selected two different input patches with 8 and 16 pixels. From the linear evaluation, the authors mention the lack of significant improvement between methods. The interesting results come in the fine-tuning performance given different label ratios. They observe that self-supervised models still perform well when the number of annotations is reduced. This aspect is critical as expert labels are time-consuming to acquire. More specifically,

ViT-based architectures achieve the best performances.

Out of the most recent SSL models applied to digital slides, we cite HIPT [29] and SRCL [147] that both use transformers as encoders. In HIPT, they use DINO-like architectures to learn hierarchical features. This approach is interesting as it tackles the problem of image representation at different magnifications. The authors introduce a simple way to aggregate multi-scale local features to generate a WSI-level representation. This work is a first step toward an efficient patient data representation of WSIs. One can imagine using such technique to perform image retrieval given rare pathologies. The second work, SRCL, focuses on positive pair mining to improve model performance on various downstream tasks. It is one of the few latest works that use contrastive learning.

2.5 Survival Analysis

Survival analysis is defined as all the techniques used to estimate entities' expected survival times and hazards. It can be applied to various domains, such as failure in mechanical systems or death of living organisms in life sciences. Here, we focus on survival analysis for the medical applications where we aim to model the time-to-event (TTE). The TTE is typically set as the time interval between the disease diagnosis (or surgery) and the event's occurrence (or loss of patient follow-up). We define three main TTE targets as overall survival (OS), disease-free survival (DFS), and disease-specific survival (DSS) which are defined as:

- **OS**: Overall survival time of a given patient. The occurrence of the event is considered as the death of the patient. The cause of death is not necessarily linked to the tumor. It can include organ failure, premature death, or postoperative death,
- **DFS**: Survival time without recurrence/relapse of the disease. The occurrence of the event is considered as the recurrence of the cancer,
- **DSS**: Survival time with respect to a specific disease. The occurrence of the event is considered as the death of the patient from a given disease (*e.g.* CRC). This endpoint is more difficult to assess as the patient's death is often not directly related to the tumor (*e.g.* organ failure, pneumonia, or ascites).

To account for postoperative death (*i.e.* patient dying following the surgery), it is recommended to discard the set of patients where the event occurs less than 30-day after surgery [74] (or less than 90-day for a safer margin [39]). Regarding the patient follow-up time, the gold standard is to consider a 5-year survival time [125] (60 months). Moreover, patients with survival times longer than the 5-year standard are set to an identical time point (*e.g.* 60 + 1 months). The event entries are all forced to "no event" as the event did not occur within the 5-year interval.

During a study, losing track of the patient's status is frequent. In this case, we keep the last follow-up time as the reference and call the case "right-censored". Right censoring means we know the time is greater than the final measure, but we do not know by how much.

More formally, let us denote as $(t_i,\ e_i,\ \mathbf{x}_i)$ the data from the i-th patient of a study, $i \in \{1 \dots N\}$, where $t_i \in \mathbb{R}_+$ is the TTE, $e_i \in \{0, 1\}$ the event status, and $\mathbf{x}_i \in \mathbb{R}^K$ a set of K patient features. For the events, we denote as 1 the event's occurrence and as 0 the right-censored event. In the next part of the section, we present well-known tools used to assess survival time. We recommend the use of Python packages for data analysis [115, 40].

We first present the non-parametric Kaplan-Meier (KM) survival estimator in subsection 2.5.1. We then move to the parametric Cox proportional hazards (CPH) approach to compute hazard rates of clinical features in subsection 2.5.2. Then, we discuss the use of univariate and multivariate models and how to ensure their statistical significance subsection 2.5.3. Finally, we introduce useful metrics to assess the prediction capability models subsection 2.5.4.

2.5.1 Kaplan-Meier Estimator

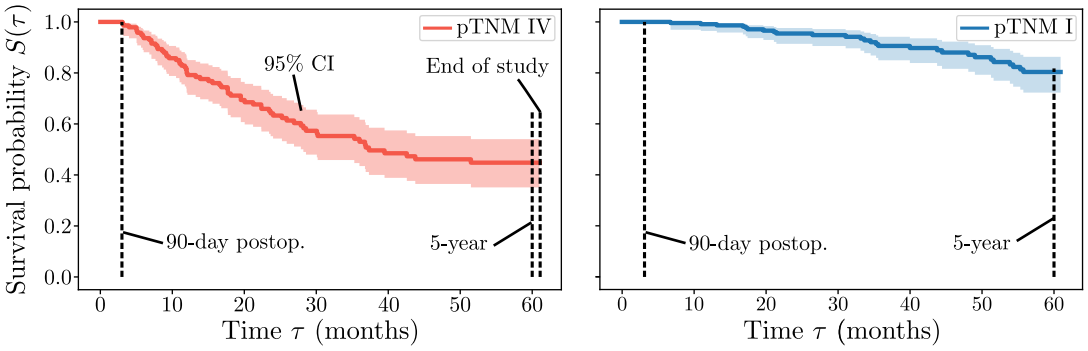
The KM estimator [76] is the most common non-parametric estimator of survival functions. It aims at predicting the probability of a patient surviving past a specific time τ and is given as:

$$S(\tau) = \prod_{\{j \,|\, t_j \leq \tau\}} \left(1 - \frac{d_j}{n_j}\right), \tag{2.44}$$

$$d_j = \sum_{i=1}^{N} \mathbb{1}_{(t_j = t_i)} \mathbb{1}_{(e_i = 1)}, \qquad \text{and} \qquad n_j = \sum_{i=1}^{N} \mathbb{1}_{(t_i \geq t_j)},$$

where $\mathbb{1}$ is the indicator function. For each time step $t_j \leq \tau$, we compute the number of deaths occurring (d_j) and the number of patients left in the study (n_j) prior to time τ. The final survival probability at time τ is the cumulative product of the current and previous estimates. To compute the confidence interval (CI), we estimate the variance using Greenwood's [58, 102] approach:

$$\mathrm{Var}(S(\tau)) = S(\tau)^2 \sum_{\{j \,|\, t_j \leq \tau\}} \frac{d_j}{n_j(n_j - d_j)}. \tag{2.45}$$



At risk	142	119	80	56	42	35	28	230	218	179	148	110	93	77
Censored	0	3	21	32	38	43	49	0	11	44	72	104	116	126
Events	0	20	41	54	62	64	65	0	1	7	10	16	21	27

Figure 2.16 – Kaplan-Meier (KM) estimator for stage I and IV CRC patients. We use a 90-day threshold for postoperative risk and consider a 5-year interval. The KM (red/blue) is displayed with a 95% confidence interval (CI) (light red/blue). Samples with survival time larger than the 5-year threshold are set to 60+1 months with no event occurrence. The table below displays the number of patients left in the study, cumulative censored samples, and cumulative events for given time steps.

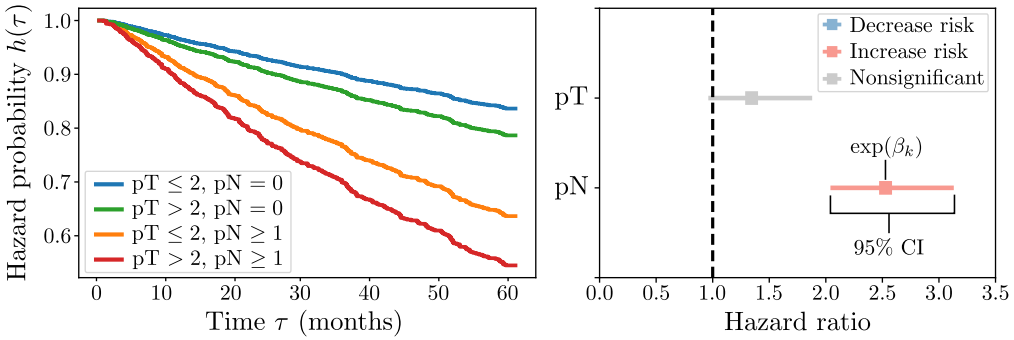
Considering the approximation of the 95% CI (*i.e.* $\simeq 2$ times the standard deviation), the final function boundaries are given as $S(\tau) \pm 2\sqrt{\text{Var}(S(\tau))}$. An example of the KM estimator applied to stages I and IV CRC patients is given in Figure 2.16. We can observe a lower survival probability after the 5-year period for the group with stage IV cancer compared to stage I.

2.5.2 Cox Proportional Hazards

The Cox proportional hazards (CPH) is a regression model predicting proportional hazards from covariates. Survival functions are used to model the occurrence of specific events given time τ. Proportional hazards, on the other hand, focus on the influence of patients' features. For example, given a set of patients with metastasis (pM1) and patients without metastasis (pM0), we are interested in knowing the hazard ratio (HR) between the two groups. Moreover, the proportional hazards are not limited to categorical variables as opposed to KM estimator. The hazard function h of a patient with features $\mathbf{x}$ is defined as:

$$h(\tau \mid \mathbf{x}) = \overbrace{h_0(\tau)}^{\text{baseline hazard}} \underbrace{\exp \overbrace{\sum_{j=1}^{K} (x_j - \overline{x}_j)\beta_j}^{\text{log-partial hazard}}}_{\text{partial hazard}} . \tag{2.46}$$

It comprises a baseline function $h_0(\tau)$ that is common to all patients in the study and a



(a) Hazard function (b) Multivariate HR

Figure 2.17 – Cox proportional hazards (CPH) regression for tissue invasion depth (pT) and positive lymph nodes (pN). (a) Hazard function for different clinical features states. The highest risk occurs with both high pN (pN > 0) and pT (pT > 2). (b) The hazard ratios (HRs) for pT and pN. Only pN is statistically significant and increases the risk when considering the 95% confidence interval (CI).

partial hazard term. As the function uses an exponential term to estimate hazards, we refer to the inner terms as log-partial hazards. For each feature x_j, we estimate its mean over the population as $\overline{x}_j$ (*i.e.* mean of the feature over patients). The parameters β_j are regressed and express the HRs.

The regression of the parameters and the definition of the CIs are based on partial log-likelihood (LL) maximization, which is out of the scope of this document. We recommend reading work on untied times [16] and tied times [49] for additional information about the topic. Note that in the presence of many variables, it is ordinary to add a regularization term on the parameters using l_1-norm. An example of CPH regression to tissue depth invasion (pT) and positive lymph nodes (pN) is depicted in Figure 2.17. We show the multivariate HR with CI. When plotting the hazard ratios, if a parameter is located on the right of the baseline (*i.e.* HR > 1), we consider it increases the patient risk. On the contrary, if the parameter is located on the left (*i.e.* HR < 1), then the risk is decreased. Finally, if the confidence interval crosses the baseline (*i.e.* HR $= 1$), then the variable is statistically nonsignificant.

2.5.3 Forward Selection

When performing survival analysis, we aim to select the variables that better explain the patient survival and hazard functions. Univariate models check whether a single variable is linked to the patient outcome. In the multivariate setting, we study how combining multiple variables influences survival.

In the multivariate case, it is essential to check for variable independence. Two features may be highly correlated and then redundant to the model fitting. Here, we propose

to use forward selection and check for variable significance. Let's assume we have two multivariate models h_1, h_2:

$$\begin{aligned} h_1(\tau \mid \mathbf{x}) &= h_0(\tau) \exp(\sum_{j=1}^{K} (x_j - \overline{x}_j)\beta_j), \\ h_2(\tau \mid \mathbf{x}) &= h_0(\tau) \exp((x_{K+1} - \overline{x}_{K+1})\beta_{K+1} + \sum_{j=1}^{K} (x_j - \overline{x}_j)\beta_j), \end{aligned} \tag{2.47}$$

We consider the first model as the reference model. The second model is the same as the first model except for the fact that it includes one additional variable and its associated parameter (*i.e.* x_{K+1} and β_{K+1}). We want to know whether adding a new variable benefits the reference model. To do so, we define the log-likelihood ratio test (LLRT):

$$\chi^2_{LLR} = -2(\mathrm{LL}(h_1) - \mathrm{LL}(h_2)), \tag{2.48}$$

where LL is the maximized log-likelihood under each model [91]. We use the χ^2 test with one degree of freedom ($df = (K+1) - K = 1$) to check for statistical significance as a single variable is added to the base model.

In practice, when dealing with multiple variables, we first perform an univariate analysis for each of the K variables. Out of the tested variables, we keep the ones where the p-value is lower than a certain threshold (*e.g.* $p < 0.1$). We are then left with $K' < K$ variables. The use of 0.1 as a threshold for confidence value is empirical and subject to discussion. The main idea behind this threshold is to keep the variables that are (nearly) significant when fitting the univariate models (*i.e.* threshold slightly above $p = 0.05$). As a second step, we select one of the K' univariate models that maximize the LL as the reference model. We then try to add to the reference model a second variable out of the $K' - 1$ left. To do so, we test all the possible pairs and keep the one that maximizes the LL. Finally, we use LLRT to assess whether there is a benefit in adding the newly selected variable. The process is then iteratively repeated until either no variables left or LLRT fails.

It is important to note that a variable can achieve statistical significance when used in a univariate model but fails in a multivariate one. The interpretation is that there is another variable in the multivariate model that captures the information of that variable. We, therefore, have redundancy of the data. A simple example would be using pT and pTMN in a multivariate model. Both variables are highly correlated since pT information is encapsulated in the pTNM classification. As a result, there is no benefit in considering both variables for model fitting.

2.5.4 Metrics

With the previously presented approaches, we can now predict survival and hazard probabilities given clinical data. However, we lack a metric to compare the quality of the predictors. In this section, we provide additional tools to assess the discriminating power of a survival predictor.

Concordance Index

Let be a set of N patients where (t_i, e_i, x_i) are the data of the i-th patient, $t_i \in \mathbb{R}_+$ is the TTE, $e_i \in \{0, 1\}$ the event status, and $x_i \in \mathbb{R}$ a predicted score. The concordance index (C-Index) aims to assess the quality of the predicted survival scores. To do so, we want to check how many patient pairs are properly ordered. To check for order, we consider pairs (i, j) where $t_i \leq t_j$. We define the set of admissible pairs:

$$\mathcal{C} = \{(i, j) \mid t_i \leq t_j, e_i = 1\} \qquad i, j \in \{1, \ldots, N\}, \tag{2.49}$$

where N is the total number of patients. Here, we cannot order right-censored samples. The reason is that even though we know that for a right-censored patient i, the TTE is at least t_i, we do not have any information about the upper bound. As a result, we only order samples based on the occurrence of events (*i.e.* $e_i = 1$). The C-Index is given as:

$$\text{C-Index} = \frac{1}{|\mathcal{C}|} \sum_{i,j \in \mathcal{C}} \mathbb{1}_{(t_i < t_j)} \mathbb{1}_{(x_i \leq x_j)} + \frac{1}{2} \mathbb{1}_{(t_i = t_j)}. \tag{2.50}$$

The final metric lies in the interval $[0, 1]$ where 0 means perfect anti-concordance, 0.5 random predictions, and 1 perfect concordance. It is impossible to order tied events (*i.e.* occurring simultaneously). Consequently, tied events are set to the default values (*i.e.* 0.5) to avoid any influence on the metric. As a result, the C-Index index can be seen as an estimation of the percentage of event pairs that are correctly ordered.

Brier Score

The C-Index is a ranking metric that does not consider the correlation of the variable to the time-to-event (TTE). Ideally, both the predicted metric and the TTE should follow the same trend.

Let's assume that the feature x_i indicates the probability of i-th patient to be even free at time t_i. We introduce the Brier score (BS) [17] as the mean squared error function

applied to survival forecasting. It is defined for the univariate case as:

$$\mathrm{BS} = \frac{1}{N}\sum_{i=1}^{N}(t_i - x_i)^2. \tag{2.51}$$

However, this definition assumes both variables are normalized. Moreover, it does not consider the right-censoring aspect of the data. A time-dependent generalization of the metric to right-censored data is proposed [57]:

$$\mathrm{BS}(\tau) = \frac{1}{N}\sum_{i=1}^{N} \underbrace{\mathbb{1}_{(t_i \leq \tau)}\mathbb{1}_{(e_i=1)}\frac{(0-x_i)^2}{\hat{G}(t_i)}}_{\text{Event already occurred}} + \underbrace{\mathbb{1}_{(t_i>\tau)}\frac{(1-x_i)^2}{\hat{G}(\tau)}}_{\text{No event yet}}, \tag{2.52}$$

where $\hat{G}(t)$ is the probability of an event occurring after time t. Finally, the integrated Brier score (IBS) is given as the integration of all time contributions (*i.e.* time-weighted average):

$$\mathrm{IBS} = \int \mathrm{BS}(\tau)\, d\tau. \tag{2.53}$$

2.6 Dataset

Throughout this work, we use multiple datasets to train our models and perform survival analysis. We first introduce the set of images used for classification and segmentation tasks in subsection 2.6.1. We then give an overview of the clinical data used for survival analysis in subsection 2.6.2.

2.6.1 Classification and Segmentation

We use multiple external datasets, such as Kather 16 (K16), Kather 19 (K19), colorectal cancer tissue phenotype (CRCTP), and SemiCol challenge. All datasets contain labeled patches extracted from HE-stained WSIs of different tissue types found in the human gastrointestinal tract. We also use multiple in-house cohorts to train our self-supervised models. An overview of all the dataset information is presented in Table 2.3. We report the status of the data (public or private), the number of classes, tiles, and WSIs, the size and magnification of the patches, and the scanner's resolution at the given magnification.

Throughout this work, we try to have consistency in class definitions. Hence we define 11 different base classes in this work as advent (ADV), adipose (ADI), background (BACK), blood (BLOOD), complex stroma (CSTR), debris (DEB), lymphocytes (LYM), mucin

Table 2.3 – List of the main CRC datasets used for training along with their information. We report the status of the datasets (public or private), the number of available classes, tiles, and WSI, as well as the tiles' size and target magnification. We also include information about the scanner resolution.

Datasets	Status	Classes	Tiles	Slides	Size	Magn.	Res.
K16 [79]	Public†	8	5,000	10	150 px	20×	0.495 µm/px
K19 [77]							
Train	Public†	9	100,000	86	224 px	20×	0.5 µm/px
Val	Public†	9	7,180	50	224 px	20×	0.5 µm/px
CRCTP [71]							
Train	Public†,‡	7	196,000	14	150 px	20×	NA
Test	Public†,‡	7	85,000	6	150 px	20×	NA
In-House [1, 2, 4]							
$\mathcal{D}_{\text{DNR}}$	Private	-	650,000	660	224 px	20×	0.486 µm/px
$\mathcal{D}_{\text{SRMA-WSI}}$	Private	-	199,500	665	448 px	40×	0.243 µm/px
$\mathcal{D}_{\text{SRMA-ROI}}$	Private	9	3	3	NA§	2.5×	3.885 µm/px
$\mathcal{D}_{\text{C2R-WSI}}$	Private	9	270,000	665	512 px	10×	0.971 µm/px
$\mathcal{D}_{\text{C2R-ROI}}$	Private	2	14	7	NA§	20×	0.486 µm/px
SemiCol	Private†	10	1,759	20	3000 px	20×	0.5 µm/px

† Last checked on 14 Aug 2023.
‡ Use of the 2nd fold that ensures patient-level separation between training ans testing.
§ Not applicable as no fixed sizes. Representation of WSIs large area.

(MUC), muscle (MUS), normal mucosa (NORM), stroma (STR), and tumor (TUM). The class definitions are presented in Table 2.4.

K16 Dataset

The K16 dataset [79] contains $5,000$ patches (150px × 150px, 74µm × 74µm) from multiple HE stained WSIs from the UMM pathology archive (University Medical Center Mannheim, Mannheim, Germany). All images are digitized using a scanner magnification of 20× (0.495µm/px). There are eight classes of tissue phenotypes, namely tumor epithelium, simple stroma (homogeneous composition and smooth muscle), complex stroma (stroma containing single tumor cells and/or few immune cells), immune cells, debris (including necrosis, erythrocytes, and mucus), normal mucosal glands, adipose tissue, and background (no tissue). The dataset is balanced with 625 patches per class and is publicly available online[1].

K19 Dataset

The K19 dataset [78] consists of patches depicting nine different tissue types: tumor tissue, stroma, normal colon mucosa, adipose tissue, lymphocytes, mucus, smooth muscle, debris,

[1] `https://zenodo.org/record/53169`.

Table 2.4 – Definition and abbreviation of the main classes used in our work. The definitions might differ slightly based on the application.

Class	Abrev.	Definition [78, 79, 93, 101]
Advent	ADV	Adventitial tissue, pericolic fat tissue, including large vessels.
Adipose	ADI	Also known as fat tissue, it is a connective tissue composed of adipocytes. Located in the outer layer of the colon.
Background	BACK	Empty tiles without the presence of tissue and slide artifacts.
Blood	BLOOD	Erythrocytes without any stromal or other tissue.
Complex stroma	CSTR	Also referred to as tumor-associated stroma (TA-STR). Storma surrounding tumor epithelium.
Debris	DEB	Cells that underwent apoptosis or necrosis.
Lymphocytes	LYM	Immune cells. Usually extracted from lymphoid aggregates.
Mucin	MUC	Content secreted by the intestinal glands.
Muscle	MUS	Composed of the muscularis propria (circular and longitudinal). Muscularis mucosa is not included.
Normal mucosa	NORM	Normal/healthy tissue that lines the digestive tract.
Stroma	STR	Connective tissue that includes submucosal tissue and collagen.
Tumor	TUM	Composed of colorectal adenocarcinoma epithelium.

and background. The data come from the NCT Biobank (National Center for Tumor Diseases, Heidelberg, Germany) and the UMM pathology archive. Each class is roughly equally represented in the dataset. In total, there are $100,000$ patches (224px $\times$ 224px, 112µm $\times$ 112µm) in the training set. All images are digitized using a scanner at a magnification of 20$\times$ (0.5µm/px). The authors released two versions of the training set. The first includes original WSI tiles, and the second normalized samples using the Macenko approach. The validation set is composed of $7,180$ patches with unbalanced classes. The images of the validation set are all normalized following the same procedure as the training set. The data are publicly available online[2].

CRCTP Dataset

The CRCTP [71] dataset contains a total of $281,000$ patches depicting seven different tissue phenotypes (tumor, inflammatory, stroma, complex stroma, necrotic, benign, and smooth muscle) split into training and testing sets from the UHCW (University Hospitals Coventry and Warwickshire, United Kingdom). The different phenotypes are roughly equally represented in both sets. In the training set, for tumor, complex stroma, stroma, and smooth muscle, there are 35,000 patches per class (15,000 for the testing set); for benign and inflammatory, there are 21,000 (9,000 for the testing set); and for debris, there are 14,000 (6,000 for the testing set). The patches (150px $\times$ 150px) are extracted at 20$\times$ resolution from 20 HE WSIs, each one coming from a different patient. For each class, only subsets of the WSIs are used to extract the patches. The annotations are made by two expert pathologists. The author released a version of their dataset where

[2] `https://zenodo.org/record/1214456`.

they ensure data independence between training and testing data (*i.e.* data from the training set and test set comes from different WSIs). The data are publicly available online[3].

While working with the CRCTP, we found discrepancies between the data labels and the tissue images. It appears that a non-negligible part of the released data is mislabeled. Moreover, we identify multiple tiles that appear in both train and test sets. We recommend that future users consider these discrepancies and read the supplementary material in section A.1.

In-house Dataset

During this thesis, we use five in-house private datasets extracted from the archive of the IGMP (Institute of Tissue Medicine and Pathology, University of Bern, Switzerland). The datasets are named after the method they were used for. The datasets are described in detail in their corresponding chapter as they are task-specific. Please refer to chapter 3, chapter 4, and chapter 5 for more details.

SemiCol Challenge

The SemiCol dataset is proposed by the European Society of Digital and Integrative Pathology (ESDIP) as a challenge. The train set includes $1,759$ tiles (3000px $\times$ 3000px) extracted from 20 CRC WSIs from the University Hospital Cologne (Cologne, Germany) and the LMU (University Hospital of Munich, Germany). The tiles are partially labeled by expert pathologists. Both institutes use different scanners but with an equal resolution of 0.5µm/px (20×). The tiles are relatively large compared to the previously presented datasets, which gives more contextual information. There are ten different classes: tumor, normal mucosa, tumor-associated stroma (complex stroma), submucosal tissue and fat tissue, muscles, lymphocytes, necrosis, mucin, blood, and background. The authors plan to publicly release the data by early 2024[4].

Note that the challenge also includes additional data from patient biopsies. The slides are labeled as either benign or tumor. For more information, please refer to the original website.

Discrepancies in Class Definitions Between Datasets

The class definitions are not homogeneous across the datasets. Moreover, datasets do not contain the same number of tissue classes. Here, we discuss the main adaptation

[3] `https://warwick.ac.uk/fac/cross_fac/tia/data/crc-tp`.
[4] `https://www.semicol.org/data/`.

made to ensure correspondence between the datasets.

Following a discussion with expert pathologists, we group stroma/muscle and debris/-mucus as stroma and debris, respectively, to create a corresponding adaptation between K19 and K16. Moreover, the complex stroma class definition between K16 and CRCTP is not identical. The CRCTP complex stroma class contains tiles from the tumor border region and is more consistent with the tumor class in the K16 and K19 datasets. In K16, the complex stroma is not limited to the tumor border surroundings. It is defined as the desmoplastic reaction area, usually composed of a mixture of debris, lymphocytes, single tumor cells, and tumor cell clusters.

Finally, in the SemiCol dataset, the adipose is merged with the submucosal tissue and labeled as advent (ADV). When adapting other datasets to the SemiCol tiles, we neglect the submucosal part and only consider the remaining classes.

A visual representation of the occurrence and relationship of different tissue types across all datasets is available in the supplementary material (section A.2).

2.6.2 Clinical

To perform our clinical analysis we rely on $2,054$ WSIs from $1,695$ unique patients from five different cohorts named $\mathcal{P}_A$, $\mathcal{P}_B$, $\mathcal{P}_C$, $\mathcal{P}_D$, and $\mathcal{P}_E$. The main characteristics of the cohort are listed in Table 2.5. All five sets are composed of patients where either OS or DFS data are available. We excluded entries where the patients underwent preoperative treatment. We consider a 90-day postoperative threshold.

The data from $\mathcal{P}_{A,B}$ are obtained from the IGMP. Patients of $\mathcal{P}_C$ come from the Radboud UMC (University Medical Center, Nijmegen, Netherlands). Data of $\mathcal{P}_D$ comes from the cancer genome atlas (TCGA) online cohort. More specifically, from the colon adenocarcinoma (COAD) and rectal adenocarcinoma (READ) subsets where we select diagnosis slides [14] (*i.e.* including the term "DX" in the filename). TCGA does not include DFS. Finally, the data in $\mathcal{P}_E$ are solely composed of patients with stage II CRC from Mount Sinai Hospital (Toronto, Canada). Note that the histological type of the tumor (*i.e.* adenocarcinoma or mucinous) is not available for this set.

To estimate the median follow-up time, we reverse the event occurrence to focus on the loss of follow-up time as a new variable [126]. Using KM we can estimate the median time by setting a threshold at 50%. It is challenging to maintain good follow-up data over the years. In our cohorts, we observe that, on average, we tend to lose half of the patient follow-up after around 5-year. The $\mathcal{P}_B$ cohort is composed of recent patients. As a result, we lack DFS events for a large part of the cohort. For additional information about other clinical variables and the restriction of the data to stage II, please refer to the supplementary material in section A.3.

2.7 Conclusion

We conclude this chapter by summarizing the key information. Digital pathology is the study of disease and tissue based on digitized images. These broadly available digital slides, called WSIs, are tissue snapshots that depict the complex interaction between cancer and normal cells. However, their data abundance comes with a main drawback: the absence or scarcity of labels. As a result, standard supervised machine learning cannot fully benefit from all the accessible data.

To this end, we aim to take advantage of SSL to learn tissue representation without annotation. SSL allows architectures to create pre-trained models that produce good feature descriptors for downstream tasks such as classification, segmentation, and survival analysis. The use of pre-trained models typically requires fewer annotations, thus partially solving the problem of label scarcity.

In the following chapters, we tackle recurrent problems with SSL as the scarcity of labels for downstream tasks, the inherent domain shift of data, the coarse resolution of classification models, and the design of automated clinical metrics for survival analysis.

Table 2.5 – Patient cohorts with main clinical variables. For the clinical applications, we use data from five different cohorts.

Characteristics	Bern ($\mathcal{P}_A$)	Bern MSI ($\mathcal{P}_B$)	Nijmegen ($\mathcal{P}_C$)	TCGA ($\mathcal{P}_D$)	Toronto ($\mathcal{P}_E$)	All ($\mathcal{P}_{A-E}$)
Patients	383	174	556	463	118	1694
Slides	739	174	556	469	118	2054
Sex (%)						
Male	230 (60.1%)	91 (52.3%)	269 (48.4%)	237 (51.2%)	68 (57.6%)	895 (52.8%)
Female	153 (39.9%)	83 (47.7%)	287 (51.6%)	226 (48.8%)	50 (42.2%)	799 (47.2%)
T-stage (%)						
T1	20 (5.2%)	9 (5.2%)	12 (2.2%)	14 (3.0%)	-	55 (3.3%)
T2	52 (13.6%)	25 (14.5%)	82 (14.7%)	85 (18.4%)	-	244 (14.4%)
T3	214 (56.0%)	72 (41.6%)	340 (61.2%)	311 (67.3%)	101 (86.3%)	1038 (61.5%)
T4	96 (25.1%)	67 (38.7%)	122 (21.9%)	52 (11.3%)	16 (13.7%)	353 (20.9%)
N-stage (%)						
N0	198 (52.8%)	105 (63.6%)	321 (57.7%)	266 (57.6%)	118 (100%)	1008 (60.1%)
N1	102 (27.2%)	32 (19.4%)	151 (27.2%)	113 (24.5%)	-	398 (23.7%)
N2	75 (20.0%)	28 (17.0%)	84 (15.1%)	83 (18.0%)	-	270 (16.1%)
TNM (%)						
I	62 (16.2%)	26 (14.9%)	75 (13.5%)	82 (18.3%)	-	245 (14.6%)
II	134 (35.0%)	79 (45.4%)	241 (43.3%)	170 (38.0%)	118 (100%)	742 (44.2%)
III	127 (33.2%)	59 (33.9%)	218 (39.2%)	135 (30.2%)	-	539 (32.1%)
IV	60 (15.7%)	10 (5.7%)	22 (4.0%)	60 (13.4%)	-	152 (9.1%)
Histopathologic type (%)						
Adenocarcinoma	328 (86.5%)	139 (83.2%)	419 (75.6%)	383 (86.5%)	NA	1269 (82.2%)
Mucinous	51 (13.5%)	28 (16.8%)	135 ($24.4^{\%}$)	60 (13.5%)	NA	274 (17.8%)
OS (%)						
Alive	275 (72.0%)	58 (75.3%)	422 (75.9%)	375 (81.0%)	101 (85.6%)	1231 (77.1%)
Dead	107 (28.0%)	19 (24.7%)	134 (24.1%)	88 (19.0%)	17 (14.4%)	365 (22.9%)
DFS (%)						
Free	275 (86.5%)	27 (90.0%)	446 (81.8%)	NA	98 (83.8%)	846 (83.8%)
Recurrence	43 (13.5%)	3 (10.0%)	110 (18.2%)	NA	19 (16.2%)	165 (16.2%)
OS 5-year	67.5%	70.1%	71.8%	62.1%	84.8%	70.6%
OS Median follow-up (CI)	95 (89 - 101)	59 (38 - 61)	88 (79 - 108)	26 (25 - 29)	64 (62 - 67)	60 (57 - 63)
DFS 5-year	82.9%	NA	78.3%	NA	82.8%	80.0%
DFS Median follow-up (CI)	40 (38-43)	NA	67 (58 - 76)	NA	63 (60 - 65)	54 (50 - 58)

Abbreviations and definitions: Not available or few samples (NA), confidence interval (CI), and median follow-up time in months.

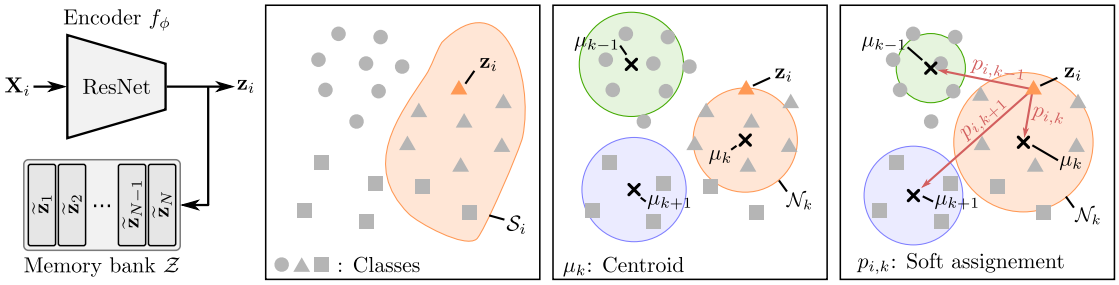
3 Divide-and-Rule: Learning from Digital Slides Structure

> Comment le haut peut diviser le bas? En créant le concept de communautés.
>
> *Diviser pour mieux régner, Horizon vertical*
> Valentin Le Du

Histopathology is characterized by its large data availability. Resected tissues from patients undergoing surgery are selected, embedded, and scanned for diagnosis. Over the years, clinical institutes have accumulated thousands of whole slide images (WSIs). The availability of such a large amount of data is a boon for large machine-learning models that tend to be more data-greedy. However, access to these large cohorts comes with a main drawback: labels' scarcity. Annotations are expensive and time-consuming to acquire for expert pathologists.

Consequently, novel methods need to find a way to learn tissue structure in an unsupervised way or to take advantage of already existing publicly available labels. Some works use, for example, unsupervised clustering [103], registered data from multiple stains [43], or manual feature extraction [53] to account from the lack of labels. Other works take advantage of the availability of patient survival data to build train end-to-end architectures [166, 90]. However, they heavily relied on noisy and compressed features from pre-trained networks that lack interpretability. Another solution is using self-supervised learning (SSL) methods [64, 167, 31] that do not require any prior annotations.

In this chapter, we propose our novel approach to learn histopathological patterns through self-supervised learning [4]. We first introduce three reference baselines in section section 3.1. Then, motivated by the advantages and pitfalls of the presented baseline, we present in section 3.2 our novel way to model tissue interactions using image reconstruction. The model relies on the use of an autoencoder to learn features in a self-supervised fashion, thus alleviating the need for labeled data. Moreover, we



(a) Encoder (b) DSC (c) DCA (d) DEC

Figure 3.1 – Baseline feature space optimization. (a) The used encoder architecture f_ϕ and memory bank $\mathcal{Z}$, (b) spatial consistency (DSC), (c) clustering assignment (DCA), and (d) embedding clustering (DEC). We show how each baseline take advantage of an embedding $\mathbf{z}_i = f_\phi(\mathbf{X}_i)$ to learn discriminant feature using different neighborhood sets $\mathcal{S}$ or $\mathcal{N}$.

impose constraints in the encoder latent space to further improve data representation. Finally, to validate the performance of our architecture, we take advantage of our model tissue description to predict patients' survival. To do so, we propose a way to aggregate feature representation within WSIs to create interpretable patient descriptors using spatial clustering in section 3.3. Finally, we fit our representation using data from a well-characterized patient cohort with clinicopathological data, including survival time in section 3.4. To accelerate research, we make our code and trained models publicly available on GitHub[1].

3.1 Constrain on Feature Embedding

This section presents three different baselines that impose constraints in the feature space to learn discriminant features. The embedding space can be seen as a compression space where we have access to a large amount of encapsulated information about the visual aspects and properties of the images. Before going through some specific example, let's define $\mathcal{W} = \{\mathbf{X}_i\}_{i=1}^N$ as the set of N image tiles with index i that compose a WSI. We can define the embedding of each tile as $\mathbf{z}_i = f_\phi(\mathbf{X}_i) \in \mathbb{R}^D$, where f_ϕ is an encoder with parameters ϕ and D the size of the embedding space. We now introduce different SSL losses as spatial consistency (DSC), clustering assignment (DCA), and embedding clustering (DEC) that can be applied to impose consistency in the feature space. The visual representation of the learning procedure is depicted in Figure 3.1.

[1] `https://github.com/christianabbet/DnR`.

3.1.1 Spatial Continuity (DSC)

We want to take advantage of the spatial consistency of the WSI. If we consider two tiles sampled from the same WSI, which are spatially close to each other, we can assume that they share similar visual content. We can define the set of tile index $\mathcal{S}_i$ that constitute the spatial neighborhood of $\mathbf{X}_i$ as:

$$\mathcal{S}_i = \{j \mid \|\text{coord}(\mathbf{X}_i) - \text{coord}(\mathbf{X}_j)\|_2 < \epsilon,\, i \neq j, \text{ and } \mathbf{X}_j \in \mathcal{W}\}\,, \tag{3.1}$$

where coord denotes the x and y spatial coordinates of the tile within the WSI and ϵ is a distance threshold. As a result, $\mathcal{S}_i$ is the set of tiles indexes within spatial distance ϵ of $\mathbf{X}_i$. Decreasing ϵ lowers the distance threshold and thus increases the likelihood for the two patches to share information. We define the spatial continuity loss as:

$$\begin{aligned} \min \mathcal{L}_{\text{DSC}} &= \min_{\phi} \frac{1}{N} \sum_{i=1}^{N} \frac{1}{|\mathcal{S}_i|} \sum_{j \in \mathcal{S}_i} \|\mathbf{z}_i - \mathbf{z}_j\|_2 \\ &= \min_{\phi} \frac{1}{N} \sum_{i=1}^{N} \frac{1}{|\mathcal{S}_i|} \sum_{j \in \mathcal{S}_i} \|f_\phi(\mathbf{X}_i) - f_\phi(\mathbf{X}_j)\|_2\,. \end{aligned} \tag{3.2}$$

One of the drawbacks of this approach is the definition of the optimal value of ϵ. In most cases, the selection of ϵ is class and context-dependent. If not careful, the learning process might optimize the representation of tissues that do not share any visual similarity. Moreover, the model lacks negative examples to repel features in the current setup. A trivial solution to the loss would be to set all features to a constant vector. In this case, the loss reaches a local minimum as all the feature space representation collapses.

3.1.2 Cluster Assignment (DCA)

Another way to impose consistency in the feature space is to consider a clustering approach. Here, we define a set of K clusters with centroids $\mu_k \in \mathbb{R}^D$. We assign each embedding $\mathbf{z}_i$ of our feature space to the nearest cluster $\mathcal{N}_k$ as:

$$\begin{cases} \mathcal{N}_k^{(t)} & = \{i \mid \|\mathbf{z}_i - \mu_k^{(t-1)}\|_2 < \|\mathbf{z}_i - \mu_j^{(t-1)}\|_2\,, \forall j \neq k\,, j \in [1 \dots K]\} \\ \mu_k^{(t)} & = \frac{1}{|\mathcal{N}_k^{(t)}|} \sum_{i \in \mathcal{N}_k^{(t)}} \mathbf{z}_i \end{cases}. \tag{3.3}$$

The estimation of the centroids μ_k is the average of the given cluster embeddings and changes with time steps t. For each cluster, we aim at minimizing the distance of the

embeddings to their cluster center μ_k. The definition of the optimization loss is given as follows:

$$\min \mathcal{L}_{\text{DCA}} = \min_{\phi} \frac{1}{N} \sum_{k=1}^{K} \sum_{i \in \mathcal{N}_k} \|\mathbf{z}_i - \mu_k\|_2. \tag{3.4}$$

In this setup, the frequency of the update is prone to discussion. A recent work suggests updating clusters between training epochs [104]. However, it means the embeddings must be recomputed at each epoch, thus slowing down the training time. An alternative solution is to use a memory bank $\mathcal{Z} = \{\widetilde{\mathbf{z}}_i\}_{i=1}^{N}$ as an embedding substitute. When a new embedding $\mathbf{z}_i$ is generated with the encoder, we update the memory bank with the corresponding entry:

$$\widetilde{\mathbf{z}}_i \leftarrow m\,\widetilde{\mathbf{z}}_i + (1-m)\,\mathbf{z}_i \qquad \text{and} \qquad m \in [0,1], \tag{3.5}$$

where m controls the momentum. In practice, m is kept high to avoid giving too much weight to the new samples. It ensures more stability during the learning process. The memory bank is initialized before training using a normal distribution. When computing the centroids, we use the Equation 3.3 where we substitute $\mathbf{z}_i$ with the entries $\widetilde{\mathbf{z}}_i$ of the memory bank $\mathcal{Z}$. It lets us easily update the cluster assignment and centroids at each optimization step.

Still, the cluster assignment approach raises a few concerns. The first is the initialization of the centroids. In the standard K-means, they are randomly initialized, which makes them sensitive to outliers and changes in feature space densities. Secondly, dense clusters are prioritized due to their significant contribution to the loss term. As a result, smaller clusters might eventually be emptied and collapse. To ensure a good initialization of centroids, a workaround is to run initialization with multiple restarts and select the solution that minimizes Equation 3.4. In addition, clusters are re-initialized when they collapse to avoid the degradation of the feature representation. In other words, when a small cluster is emptied, we set a new center located inside another larger, non-empty cluster. By doing so, we avoid big clusters to gather all the information.

3.1.3 Embedded Clustering (DEC)

With the previously defined DCA, we assume that the feature space is uniformly distributed and that samples represent single classes. As a result, the assignment of a cluster only depends on the nearest center, which can be prone to error. For example, an outlier will likely alter the clustering quality by significantly shifting its mean toward itself. Here, we assume that our feature space comprises a soft mixture of sparse and

dense areas. Based on this assumption we estimate a Student's t distribution [140, 152] to measure the similarity between our embedding $\mathbf{z}_i$ and centroids μ_k as:

$$q_{i,k} = \frac{(1 + \|\mathbf{z}_i - \mu_k\|_2^2)^{-\frac{1}{2}}}{\sum_{k}^{K}(1 + \|\mathbf{z}_i - \mu_k\|_2^2)^{-\frac{1}{2}}}\,. \tag{3.6}$$

We can see the variable $q_{i,k}$ as the probability of sample $\mathbf{z}_i$ to belong to the cluster k with mean μ_k. The samples are, therefore, softly assigned (*i.e.* not tied to a single cluster). The clusters are initialized at the beginning of the training following the standard K-means procedure. The second step is to define a target distribution to match. The selected function needs to strengthen predictions [152] and is defined as:

$$p_{i,k} = \frac{q_{i,k}^2 / f_k}{\sum_{k'}^{K} q_{i,k'}^2 / f_{k'}} \qquad \text{and} \qquad f_k = \sum_{i}^{N} q_{i,k}\,. \tag{3.7}$$

Raising the prediction $q_{i,k}$ to the second power sharpens the distribution and encourages high-confidence assignment. In addition, the predictions are normalized with the so-called *soft cluster frequencies* f_k. Locally dense clusters can bias the learning of the feature space due to their large amount of assigned samples. As a result, the representation of other more sparse areas is neglected. The variables f_k are inversely proportional to the cluster sizes and aim to decrease dense clusters' impact. The objective loss matches the source distribution $\mathbf{Q} = (q_{i,k})_{1 \leq i \leq N, 1 \leq k \leq K}$ to the selected target one $\mathbf{P} = (q_{i,k})_{1 \leq i \leq N, 1 \leq k \leq K}$ using Kullback-Leibler (KL) divergence as:

$$\min \mathcal{L}_{\text{DEC}} = \min_{\phi} \text{KL}(\mathbf{P} \,||\, \mathbf{Q}) = \min_{\phi} \sum_{i}^{N} \sum_{k}^{K} p_{i,k} \log \frac{p_{i,k}}{q_{i,k}}. \tag{3.8}$$

The limitation of this approach is the need for a pre-training step. The initialization of the cluster centers is performed only once at the beginning of the optimization. As a result, the model needs to first learn feature representation through another approach, such as autoencoders or stacked autoencoders [143].

3.2 Proposed Approach

This section introduces our novel Divide-and-Rule (DNR) approach. The model aims to solve the presented baselines' limitations while taking advantage of SSL. We first introduce our self-supervised transfer colorization scheme that takes advantage of staining

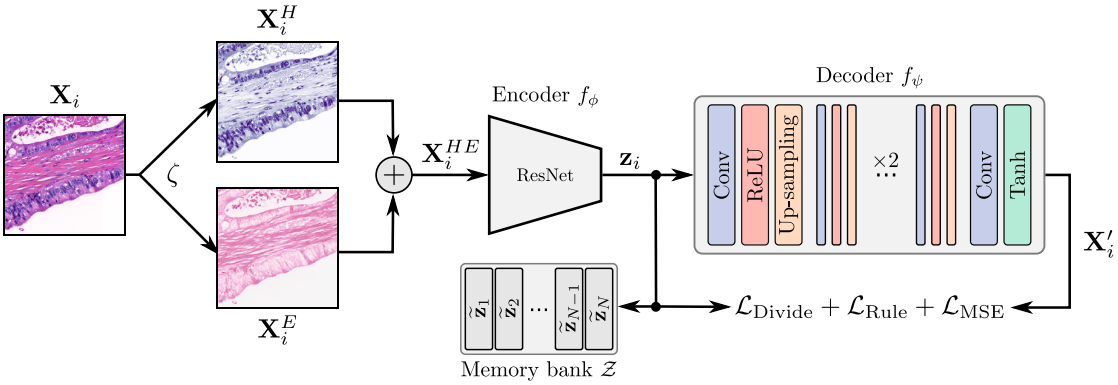


Figure 3.2 – The architecture of our proposed Divide-and-Rule (DNR) approach. The model is composed of an encoder f_ϕ and decoder f_ψ. It takes as input $\mathbf{X}_i^{HE}$ the concatenation of the hematoxylin $\mathbf{X}_i^H$ and eosin component $\mathbf{X}_i^E$ of an image $\mathbf{X}_i$ extracted using function ζ. The memory bank $\mathcal{Z}$ is updated with the embedding vector $\mathbf{z}_i$ and used to learn discriminant features through $\mathcal{L}_{\text{Divide}}$ and $\mathcal{L}_{\text{Rule}}$. The decoder reconstructs the output image $\mathbf{X}'_i$ which is used to optimize $\mathcal{L}_{\text{MSE}}$.

information in subsection 3.2.1. Then, we present our DNR loss to represent image patches based on their spatial proximity and embedding. Finally, we motivate our region of interest (ROI) detection scheme that focuses on tumor representation in subsection 3.2.3. The architecture of the presented method is depicted in Figure 3.2.

3.2.1 Learning from Staining

To learn a first representation of our embedding, we take advantage of the staining information. Let's assume we have access to N images $\mathbf{X}_i, i \in \{1, \ldots, N\}$. For each input image $\mathbf{X}_i$, we extract its hematoxylin $\mathbf{X}_i^H \in \mathbb{R}^{H \times W}$ and eosin $\mathbf{X}_i^E \in \mathbb{R}^{H \times W}$ channel using stain decomposition $\zeta : \mathbb{R}^{H \times W \times 3} \to \mathbb{R}^{H \times W \times 2}$. The resulting matrix $\mathbf{X}_i^{HE}$ is given in Equation 3.9 and is the concatenation along the channel axis of both $\mathbf{X}_i^H$ and $\mathbf{X}_i^E$ components.

$$\zeta(\mathbf{X}_i) = \mathbf{X}_i^{HE} = \begin{pmatrix} \mathbf{X}_i^H & \mathbf{X}_i^E \end{pmatrix} . \tag{3.9}$$

The image $\mathbf{X}_i^{HE}$ is fed to an encoder $f_\phi : \mathbb{R}^{H \times W \times 2} \to \mathbb{R}^D$ with parameters ϕ to generate $\mathbf{z}_i \in \mathbb{R}^D$ where D is the dimension of the embedding space. As a second step, the vector $\mathbf{z}_i$ is fed to the decoder $f_\psi : \mathbb{R}^D \to \mathbb{R}^{H \times W \times 3}$ with parameters ψ to create the reconstructed image $\mathbf{X}'_i \in \mathbb{R}^{H \times W \times 3}$. The decoder is created by alternating five convolutional layers, rectified linear unit (ReLU), and bilinear up-sampling until the desired output dimension is reached. We add a final convolution and hyperbolic tangent (Tanh) layer for the regression output. The architecture of the decoder is selected to

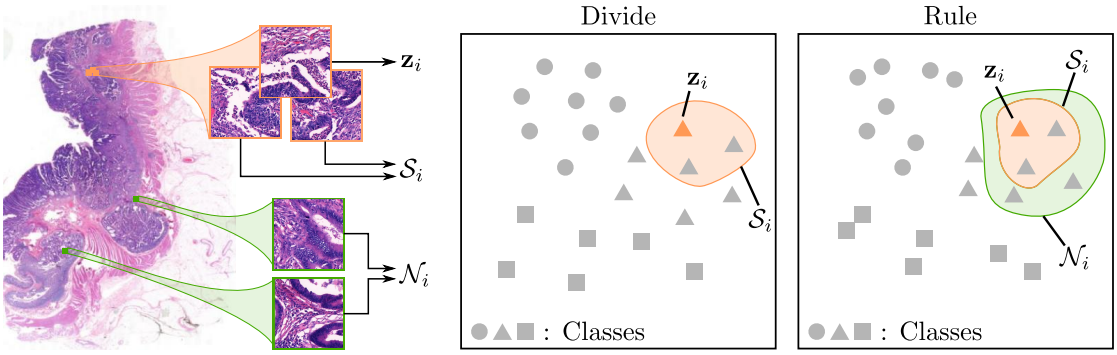


Figure 3.3 – Illustration of the optimization process of the divide and rule components. Left: the sampling of whole slide image tile. Right: the divide and rule settings. We show the example of a tile embedding $\mathbf{z}_i$, its overlapping tiles $\mathcal{S}_i$, and its feature space neighbors $\mathcal{N}_i$. All presented tiles depict tumor tissue and are assumed positive to $\mathbf{z}_i$.

have a minimal number of parameters. This approach allows us to ensure that the encoder generates a meaningful latent space, as the decoder is not complex enough to achieve perfect sample reconstructions. Finally, we define the optimization process as the minimization of the reconstruction loss between the input and predicted images:

$$\begin{aligned} \min \mathcal{L}_{\text{MSE}} &= \min_{\phi,\psi} \frac{1}{N} \sum_{i=1}^{N} \|\mathbf{X}_i - \mathbf{X}'_i\|_2 \\ &= \min_{\phi,\psi} \frac{1}{N} \sum_{i=1}^{N} \|\mathbf{X}_i - f_\psi \circ f_\phi \circ \zeta(\mathbf{X}_i))\|_2 \,. \end{aligned} \tag{3.10}$$

3.2.2 Divide-and-Rule

The principle behind our self-supervised learning approach is based on spatial proximity and entropy minimization. The overall pipeline is depicted in Figure 3.3. First, we consider that any two spatially adjacent WSI image patches (positive pairs) are more likely to share similar visual content and thus share more information in the feature space than two distant WSI patches (negative pairs). One of the limitations of spatial proximity is the definition of the distance threshold ϵ (see Equation 3.1). Here, we impose overlapping between positive patches to ensure that the positive pairs share similar histomorphological patterns. We denote the set of patch indexes that overlap with patch $\mathbf{X}_i$ as:

$$\mathcal{S}_i = \{j \mid \|\text{coord}(\mathbf{X}_i) - \text{coord}(\mathbf{X}_j)\|_2 \leq \frac{1}{2} \min(H, W),\, i \neq j\} \,, \tag{3.11}$$

where the distance is conditioned by the patch's original size. We then define a second set of positive tiles using tissue similarity in the feature space. To do so, we first create a memory bank $\{\tilde{\mathbf{z}}_i\}_{i=1}^{N} = \mathcal{Z}$ to keep track of past embedding $\mathbf{z}_i$ using momentum as in Equation 3.5. Here, we can interpret $\tilde{\mathbf{z}}_i$ as an estimation of the embedding $\mathbf{z}_i$. The memory bank allows us to query any tissue embedding during training. Moreover, we consider that $\mathbf{z}_i$ and $\widetilde{\mathbf{z}}_i$ are normalized entries $\|\mathbf{z}_i\|_2 = \|\widetilde{\mathbf{z}}_i\|_2 = 1 \; \forall i \in [1 \dots N]$. Finally, we define the index set of memory bank patches that achieve a low cosine distance when compared to the query $\mathbf{z}_i$:

$$\mathcal{N}_i = \{j \mid (1 - \widetilde{\mathbf{z}}_j^\top \mathbf{z}_i) \leq (1 - \widetilde{\mathbf{z}}_k^\top \mathbf{z}_i), \forall k,\, i \neq j\}. \tag{3.12}$$

Here, we consider the closest sample for simplicity. We assume neighbor samples in the feature space share similar visual patterns. We use cosine distance as the reference metric, which is robust in high-dimensional spaces. Another option to create $\mathcal{N}_i$ could have been to define a confidence threshold on the cosine distance. Any embedding with a lower cosine distance than the mentioned threshold would have been assigned to $\mathcal{N}_i$. However, defining a fixed threshold in practice is difficult as the cosine distance between entities evolves through the training.

We propose to follow a simple to hard learning logic to learn feature representation progressively. Given an embedding, we compute the tile's relative entropy to other tiles embedding. If a sample lies in a high-density area (*i.e.* many and close neighbors in the feature space), its relative entropy would be low. On the contrary, for a sample that lies in a small density area (*i.e.* few and distant neighbors), its entropy would be high. The entropy acts as a threshold between close and distant samples [151]. The relative entropy $\mathbf{h} = \begin{pmatrix} h_1 & \cdots & h_N \end{pmatrix} \in \mathbb{R}^N$ is defined as:

$$p_{i,j} = \frac{\exp\left(\widetilde{\mathbf{z}}_j^\top \mathbf{z}_i / \tau\right)}{\sum_{k=1}^{N} \exp\left(\widetilde{\mathbf{z}}_k^\top \mathbf{z}_i / \tau\right)}, \tag{3.13}$$

$$h_i = -\sum_{j=1}^{N} p_{i,j} \log(p_{i,j}), \tag{3.14}$$

where $\tau \in [0, 1]$ is the temperature parameter that controls the sharpness of the predictions. During the learning procedure, we split our samples into two sets $\mathcal{B}$ and $\bar{\mathcal{B}}$. Here, $\mathcal{B}$ is the set of elements that we consider to have low relative entropy, and $\bar{\mathcal{B}}$ is its complementary

set where elements have high relative entropy. They are defined as:

$$\mathcal{B} = \{i \mid h_i \text{ is top-}k \text{ in } \mathbf{h}\} \qquad \text{and} \qquad \bar{\mathcal{B}} = \{i \mid h_i \text{ is not top-}k \text{ in } \mathbf{h}\}. \tag{3.15}$$

The entries of $\mathcal{B}$ are considered easy samples as their local embedding is dense and thus should contain many relevant candidates for similarity matching. The number of top-k samples is gradually increased during training such that we go from easy samples (low entropy) to hard ones (high entropy) as:

$$k = \lfloor r\frac{e}{E} \rfloor, \tag{3.16}$$

where E is the total number of epochs, e the current epoch and $r \in]0, 1]$ a scaling factor. We define our first loss term in Equation 3.17. The objective focuses on samples with high entropy that are considered individual classes. Here, we tie together overlapping patches while maximizing the distance to other features, as the model is not confident enough to impose a constraint in the feature space.

$$\mathcal{L}_{\text{Divide}} = -\frac{1}{|\bar{\mathcal{B}}|} \sum_{i \in \bar{\mathcal{B}}} \log \left(\frac{1}{|\mathcal{S}_i|} \sum_{j \in \mathcal{S}_i} p_{i,j} \right) . \tag{3.17}$$

The second loss aims at minimizing the distance between overlapping and low entropy samples. The $\mathcal{B}$ samples are labeled as relevant candidates as they lie in a locally dense feature space. We define the optimization process as:

$$\mathcal{L}_{\text{Rule}} = -\frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \log \left(\frac{1}{|\mathcal{S}_i \cup \mathcal{N}_i|} \sum_{j \in \mathcal{S}_i \cup \mathcal{N}_i} p_{i,j} \right) . \tag{3.18}$$

The objective includes both the spatial $\mathcal{S}_i$ and feature space $\mathcal{N}_i$ constraints to improve the feature representation. The overall objective is defined as the composition of the reconstruction loss, divide, and rule loss.

$$\min_{\phi,\psi} \mathcal{L}_{\text{DNR}} = \min_{\phi} \left[\min_{\psi} \left(\mathcal{L}_{\text{MSE}} \right) + \lambda \left(\mathcal{L}_{\text{Divide}} + \mathcal{L}_{\text{Rule}} \right) \right], \tag{3.19}$$

where λ controls the importance given to the divide and rule terms. The detailed pseudocode is given in algorithm 1.

Algorithm 1: Pseudocode for DNR framework.

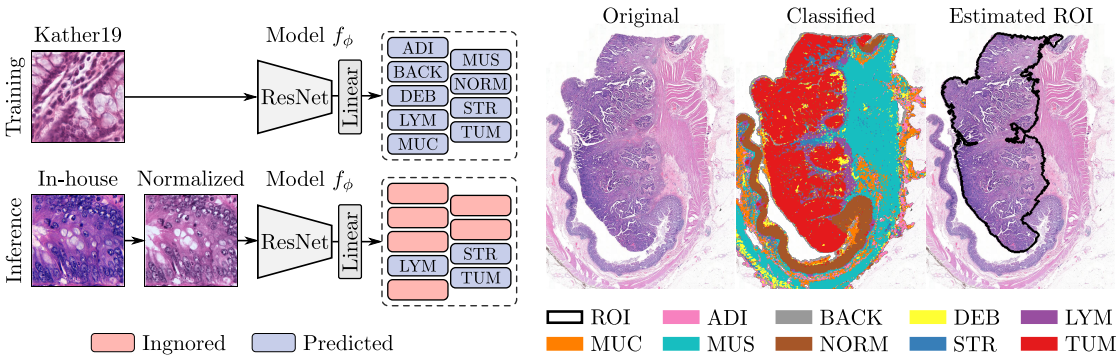
Get a set of samples $\mathcal{W} = \{\mathbf{X}_i\}_{i=1}^{N}$;
Build the set of overlapping tiles $\{\mathcal{S}_i\}_{i=1}^{N}$; ▷ Equation 3.11
Initialize memory bank $\mathcal{Z}$ by sampling from normal distribution ;
for e $= 0$ **to** $E - 1$ **do**
  Update top-k threshold ; ▷ Equation 3.16
  Compute relative entropies $\mathbf{h}$ using memory bank $\mathcal{Z}$; ▷ Equation 3.14
  Create sets $\mathcal{B}$ and $\bar{\mathcal{B}}$ based on relative entropy ; ▷ Equation 3.15
  for batch $\{\mathbf{X}_i \in \mathcal{W}\}_{i=1}^{B}$ **do**
    Extract image stain $\mathbf{X}_i^{HE} = \zeta(\mathbf{X}_i)$;
    Compute embedding $\mathbf{z}_i = f_\phi(\mathbf{X}_i)$, and reconstruction $\mathbf{X}_i' = f_\psi(\mathbf{z}_i)$;
    Normalize embeddings $\mathbf{z}_i$;
    Get the set of neighbors $\mathcal{N}_i$; ▷ Equation 3.12
    Compute loss $\mathcal{L}_{\text{Divide}}$, $\mathcal{L}_{\text{Rule}}$ and $\mathcal{L}_{\text{MSE}}$; ▷ Equation 3.19
    Optimize ϕ and ψ parameters ;
    Update memory bank $\mathcal{Z}$. ; ▷ Equation 3.5
  end
end

3.2.3 Region of Interest Detection

Our goal with the proposed DNR approach is to learn discriminative features from WSIs for survival analysis. WSIs are intricate images exhibiting diverse tissue distribution, including lymph nodes, tumor areas, and healthy tissue. Since tumor areas are the primary regions distinguishing between healthy and unhealthy patients, we presume that the cancerous region holds the most relevant information for predicting patient survival. However, WSIs often lack manual annotations regarding the presence and location of cancerous tissue. Therefore, we aim to develop an automated method for detecting tumor regions within WSIs.

To accomplish this, we use publicly available data to train a transfer learning model to classify the histological components of WSIs and identify ROIs as depicted in Figure 3.4. This approach eliminates the need for external annotations, saving time and effort. We utilize the Kather 19 (K19) dataset [77] which consists of 100,000 images of tissue from colorectal cancer (CRC) separated into nine different classes, namely adipose (ADI), background (BACK), debris (DEB), lymphocytes (LYM), mucin (MUC), muscle (MUS), normal mucosa (NORM) stroma (STR), and tumor (TUM).

Initially, we train a simple classifier $f_\theta : \mathbb{R}^{H \times W \times 3} \to \mathbb{R}^C$ with parameters θ. The model is the succession of a ResNet encoder followed by a linear projection head [66]. Here, W, H represent the height and width of the input RGB image $\mathbf{X} \in \mathbb{R}^{H \times W \times 3}$ and C is the number of classes. Subsequently, we apply our trained algorithm to WSIs using a sliding window approach. We use the stain normalization $\tau : \mathbb{R}^{H \times W \times 3} \to \mathbb{R}^{H \times W \times 3}$ to match the color space between the public data and our in-house slides to ensure that the classifier's performance is not diminished when transferring the model to different images.



(a) Training of the classification

(b) Inference on a whole slide image

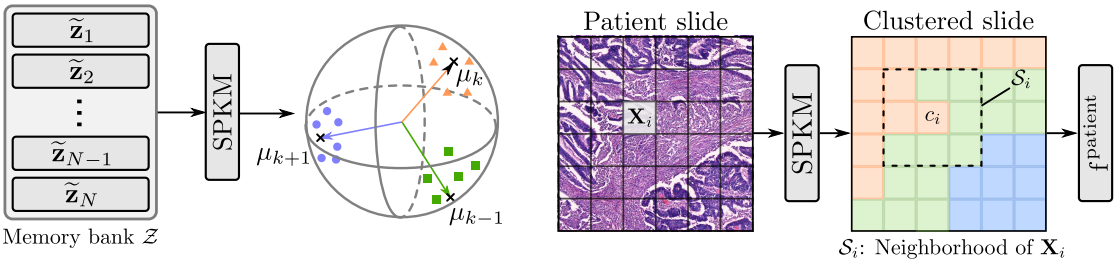
Figure 3.4 – Estimation of the region of interest (ROI) based on tumor-associated region. (a) We use a ResNet architecture f_θ trained on Kather 19 (K19) classes and use it to discriminate tumor (TUM), stroma (STR), and lymphocytes (LYM) instances on normalized in-house tiled whole slide image (WSI). (b) Overview of the classification of the WSI with the final estimated ROI highlighting the main tumor area.

We refine our approach by retraining three out of the C available classes, namely LYM, STR, and TUM, to provide an initial estimation of the tumor area and exclude a significant portion of healthy tissue regions. These classes exhibit strong discriminatory evidence for the task and have received endorsement from the pathologist. Including the TUM class is self-evident as it contains the primary tumor blobs. Moreover, lymphocytes surrounding the tumor indicate an immune reaction and are potentially linked to a higher survival score. Additionally, the intra-tumoral area is predominantly represented by the STR class. Furthermore, given the absence of information on whether the detected stroma is part of the main tumor, it is crucial to solely consider the stromal content directly attached to the tumor when creating the ROI.

3.3 Spherical Clustering

We seek to build a unique representation for each patient based on WSIs. To achieve our goal, we use a spherical K-means (SPKM) [161] approach to cluster our latent space $\mathcal{Z}$ into K different clusters with centers $\mu_k \in \mathbb{R}^D$. We assume the memory bank is a reasonable estimation of the feature space distribution. Moreover, we prefer the spherical K-means (SPKM) approach rather than a standard K-means one, as both $\mathcal{L}_{\text{Divide}}$ and $\mathcal{L}_{\text{Rule}}$ terms rely on cosine distance to compute feature similarities. As a result, the embedding representation lies on a D dimensional unit sphere.

Let's now assume that we have access to M samples $\mathbf{X}_i$, $i \in \{1, \ldots, M\}$ from a patient discriminative area (*i.e.* extracted from a ROI). To generate the patient representation, we detach the decoder f_ψ, embed every patches using the encoder f_ϕ, and assign them



(a) Fit spherical K-means (b) Build patient representation

Figure 3.5 – Overview of spherical K-means representation. (a) We use the previously learned memory bank $\mathcal{Z}$ to fit a spherical K-means (SPKM) model with K centers μ_k. (b) We apply the spherical K-means (SPKM) on whole slide image tiles $\mathbf{X}_i$ to get cluster assignment c_i and create patient final representation $\mathbf{f}^{\text{patient}}$.

to the nearest clusters to create vector $\mathbf{c} \in \mathbb{R}^M$ as:

$$\mathbf{c} = \begin{pmatrix} c_0 & \dots & c_M \end{pmatrix} \qquad \text{and} \qquad c_i = \underset{k \in \{1,\dots,K\}}{\arg\min}\ \mathbf{z}_i^\top \mu_k \,. \tag{3.20}$$

So far, we have access to a clustered representation of the WSI whose cardinality depends on the number of patches M. Unfortunately, such representation is inconsistent between patients and cannot be used as is for survival analysis. To overcome this issue, we choose to aggregate the results at the WSI level using cluster probability $\mathbf{f}^{\text{prob}} = \begin{pmatrix} \mathrm{f}_1^{\text{prob}} & \dots & \mathrm{f}_K^{\text{prob}} \end{pmatrix} \in \mathbb{R}^K$ and tile interaction as $\mathbf{f}^{\text{inter}} = \text{vec}((\mathrm{f}_{k,k'}^{\text{inter}})_{1 \le k,k' \le K}) \in \mathbb{R}^{K \cdot K}$:

$$\mathrm{f}_k^{\text{prob}} = \frac{1}{M} \sum_{i=1}^{M} \mathbb{1}_{(c_i = k)}, \tag{3.21}$$

$$\mathrm{f}_{k,k'}^{\text{inter}} = \frac{1}{M} \sum_{i=1}^{M} \frac{1}{|\mathcal{S}_i|} \sum_{j \in \mathcal{S}_i} \mathbb{1}_{(c_i = k)} \cdot \mathbb{1}_{(c_j = k')}, \tag{3.22}$$

$$\mathcal{S}_i = \{j \mid \|\text{coord}(\mathbf{X}_i) - \text{coord}(\mathbf{X}_j)\|_2 \le \frac{\sqrt{2}}{2}(W + H), i \ne j\}, \tag{3.23}$$

where $\mathcal{S}_i$ is the index set of spatial neighbor patches of $\mathbf{X}_i$ patch, W and H patches width and height, and $\mathbb{1}$ the indicator function. Here, $\mathrm{f}_k^{\text{prob}}$ denotes the probability that a patch belongs to cluster k and $\mathrm{f}_{k,k'}^{\text{inter}}$ is the probability that a patch belonging to cluster k is surrounded by elements from cluster k'. Note that the definition of the set $\mathcal{S}$ differs from the set defined in DNR approach. The overall procedure is depicted in Figure 3.5.

The concatenation of both predictors gives the final patient descriptor:

$$\mathbf{f}^{\text{patient}} = \begin{pmatrix} \mathbf{f}^{\text{prob}} & \mathbf{f}^{\text{inter}} \end{pmatrix} \in \mathbb{R}^{K(K+1)} . \tag{3.24}$$

When multiple slides are available per patient, the predictors are computed slide-wise and averaged.

3.4 Experiments

In this section, we first present the experimental setting in subsection 3.4.1. Then, we show the results of the spherical clustering in subsection 3.4.2. Finally, we perform an ablation study of our implementation and survival analysis in subsection 3.4.3.

3.4.1 Experimental Settings

For the encoder f_ϕ, we use a ResNet-18 backbone where the input layer is updated to support two input channels (hematoxylin and eosin (HE) stains). The stain estimation is performed using the Ruifrok [122] method. The latent space has dimensions $D = 512$. The decoder f_ψ is a succession of convolutional layers, ReLUs, and up-samplings (bicubic). We use Adam optimizer with $\beta = (0.9, 0.999)$ and learning rate, $lr = 1\text{e}{-3}$. The model is trained with the reconstruction loss $\mathcal{L}_{\text{MSE}}$ for 10 epochs with early stopping to create a first representation of the features with batch size $B = 32$. Then, we add $\mathcal{L}_{\text{Divide}}$ for an additional 20 epochs with $\lambda = 1\text{e}{-3}$, $r = 0.25B$, and $\tau = 0.5$. Finally, we go through 5 additional rounds using $\mathcal{L}_{\text{Rule}}$ while raising the entropy threshold between each round to refine the feature representation.

The data used to train our DNR model are generated from the patient set $\mathcal{P}_A$ which is composed of hundred slides from colorectal cancer (CRC) patients. For each WSI, we use the model f_θ trained on K19 to identify the ROI. From the ROI, we extract up to 1000 tile per slide, which brings the total of 650K individual tiles over $\mathcal{P}_A$. Tiles are normalized using the Macenko [96] algorithm to match K19 color distribution. The size of the images are 224px $\times$ 224px and their resolution is 0.486µm/px at 20$\times$. We also extract the overlapping tiles necessary for the similarity learning, which doubles the size of the training set and brings it to over 1.3M tiles. The dataset for training is named $\mathcal{D}_{\text{DNR}}$.

We compare our work with the baselines DSC, DCA, and DEC. For fair comparison, we use the ResNet-18 backbone. The number of clusters for training is tied to the one of the spherical clustering. We fit SPKM with $K = 8$ and $K = 16$ clusters to the memory bank embeddings, which we consider as a good estimation of $\mathcal{D}_{\text{DNR}}$ features.

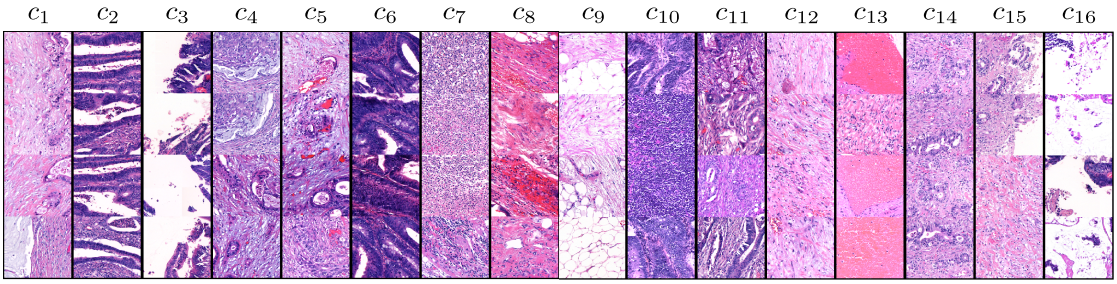


Figure 3.6 – Visualization of spherical K-means (SPKM) with $K = 16$ clusters assignation. Tiles are selected from patient set $\mathcal{P}_A$.

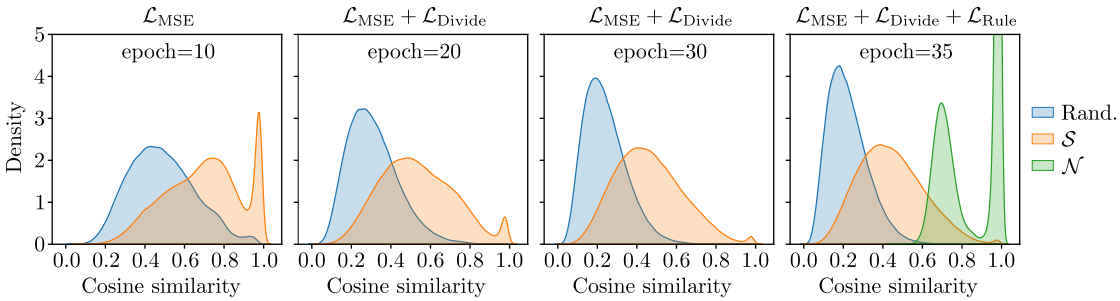


Figure 3.7 – Evolution of feature cosine similarities over the training epochs. As the training progress, the composition of losses ($\mathcal{L}_{\text{MSE}}$, $\mathcal{L}_{\text{Divide}}$, and $\mathcal{L}_{\text{Rule}}$) is updated. For a given image embedding, we compute its similarity to random, $\mathcal{S}$, and $\mathcal{N}$ entries. Similarities with respect to $\mathcal{S}$ and $\mathcal{N}$ show better correlation compared to random entries.

3.4.2 Cluster Interpretation

Examples of tile sampled from each cluster for $K = 16$ are presented in Fig. 3.6. The learning of the feature space is unsupervised and does not include prior knowledge of tissue similarities. To validate our model, we asked pathologists to analyze and evaluate the consistency of the estimated clusters. Out of the 16 clusters, we identify two clusters that highlight dense tumor areas (*i.e.* c_2, c_6). Certain clusters show tumor-to-stroma interaction as c_4, c_{11}, c_{14}. Other notable clusters focus more on inflammatory tissues (*i.e.* c_7), muscles and large vessels (*i.e.* c_8), collagen and adipose (c_9), or blood and veins (*i.e.* c_{11}). Some clusters do not directly represent the type of tissue but rather the positioning information such as c_3, which describes the edge of the WSI. The results for $K = 8$ and the baseline methods' clustering are available in supplementary section B.1.

3.4.3 Ablation Study and Survival Analysis

In this subsection, we focus on the ablation study and survival analysis. In Figure 3.7, we can observe the evolution of the cosine distance in the feature space throughout training. We start the learning procedure with the training of the autoencoder ($\mathcal{L}_{\text{MSE}}$) and then add the Divide ($\mathcal{L}_{\text{Divide}}$) and Rule ($\mathcal{L}_{\text{Rule}}$) loss terms to further improve the feature space

Table 3.1 – Multivariate survival analysis for the baselines and proposed Divide-and-Rule approach. We report losses $\mathcal{L}_{\text{MSE}}$, $\mathcal{L}_{\text{Divide}}$, and $\mathcal{L}_{\text{Rule}}$. The parameters K, N_{feat} and n denote the number of clusters, the number of features that achieve statistical relevance when performing forward selection ($p < 0.05$), and the number of patients in each set respectively. The integrated Brier score (IBS) and concordance index (C-Index) are performance indicators.

						$\mathcal{P}_A^{\text{clinical}}$ $(n = 253)^{\ddagger}$		$\mathcal{P}_A$ $(n = 374)^{\ddagger}$	
Method	$\mathcal{L}_{\text{MSE}}$	$\mathcal{L}_{\text{Divide}}$	$\mathcal{L}_{\text{Rule}}$	K	N_{feat}	IBS [57]	C-Index [63]	IBS	C-Index
Clinical					8	0.290	0.608^{***}	-	-
DSC				8	3	0.284	0.540^{+}	0.285	0.556^{**}
DCA† [104]				8	2	0.289	0.545^{**}	0.285	0.556^{***}
DEC† [152]				8	4	0.288	0.609^{**}	0.283	0.577^{**}
DNR	RGB			8	2	0.285	0.527^{**}	0.286	0.510^{***}
DNR	H&E			8	3	0.287	0.607^{*}	0.282	0.604^{***}
DNR	H&E	✓		8	3	0.283	0.595^{**}	0.284	0.592^{***}
DNR (ours)	H&E	✓	✓	8	4	0.285	0.611^{*}	0.283	0.624^{***}
DSC				16	9	0.293	0.607	0.288	0.646^{***}
DCA† [104]				16	7	0.283	0.625^{+}	0.285	0.632^{**}
DEC† [152]				16	7	**0.276**	0.641^{**}	0.276	0.643^{***}
DNR	RGB			16	0	0.290	0.500	0.290	0.500
DNR	H&E			16	5	0.282	0.636^{*}	0.280	0.632^{***}
DNR	H&E	✓		16	10	0.301	0.621^{+}	0.293	0.647^{***}
DNR	H&E	✓	✓	16	13	0.285	**0.674**	**0.273**	**0.694**

† Autoencoder is replaced with the self-supervised objective function.
‡ State of the patient cohort as in 2019 (cohort updated in 2021).
$^{+}$ $p < 0.1$; * $p < 0.05$; ** $p < 0.01$; *** $p < 0.001$ (log-rank test).

representation. We highlight the distribution of similarities between random samples in the memory bank, overlapping pairs ($\mathcal{S}$), and embedding neighborhood ($\mathcal{N}$). In the first phase (*i.e.* epochs = 10), we can already see a difference in the distributions. As the training proceeds, both random and $\mathcal{S}$ histograms tend to narrow and become more selective. When adding the final $\mathcal{L}_{\text{Rule}}$ component to our mode, we can observe that the measured similarity in the embedding space is higher. It highlights the difference in distribution between sparse embedding areas (first peak) and dense areas (far right peak). An overview of the decoder reconstruction performance is available in supplementary section B.2.

Based on the SPKM, we build our survival features on top of the predicted clusters for each patient. We report our results on the patient set $\mathcal{P}_A$ and patient subset $\mathcal{P}_A^{\text{clinical}} \subset \mathcal{P}_A$. The subset $\mathcal{P}_A^{\text{clinical}}$ is composed of patient data where all clinical metrics are available

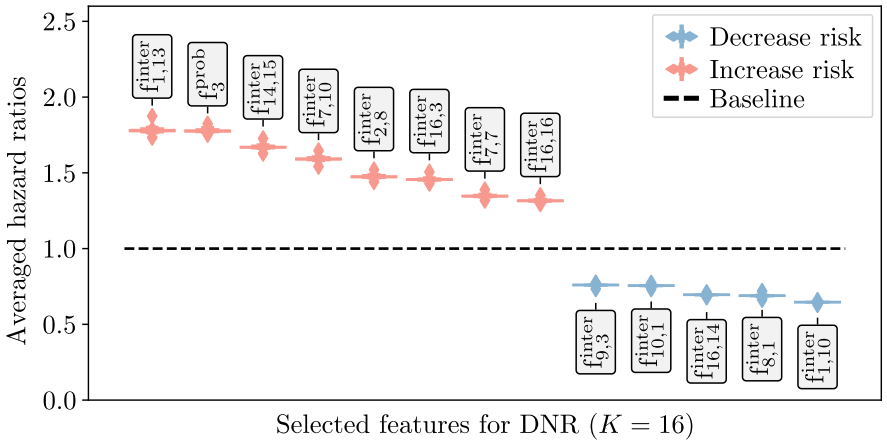


Figure 3.8 – Averaged hazard ratio over $N = 20$ runs for patient cohort $\mathcal{P}_A^{\text{clinical}}$ features (f_k^{prob} and $\text{f}_{k,k'}^{\text{inter}}$) over $K = 16$ clusters, HE staining reconstruction, divide and rule losses. The decrease and increase are depicted in blue and red, respectively.

as some clinical entries of $\mathcal{P}_A$ are missing. The generated features are fed to a Cox proportional hazards (CPH) model to predict overall survival (OS). We perform ablation studies by testing the performance of our presented architecture in different settings. As a baseline, we consider the usual clinical metrics and the approaches presented in section 3.1. To reduce the risque of overfitting, we use the leave-one-out cross validation (LOOCV) approach, where we iteratively remove one sample from the patient set and use it for validation. The results are presented in Table 3.1.

Our model outperforms previous approaches by a safe 5% margin on C-Index [63]. Regarding the autoencoder ($\mathcal{L}_{\text{MSE}}$), learning from HE representation instead of RGB help the model in finding better feature. In addition, both $\mathcal{L}_{\text{Dvidie}}$ and $\mathcal{L}_{\text{Rule}}$ terms tend to increase as well the overall performance of the model. The absence of $\mathcal{L}_{\text{Rule}}$ decreases the prediction score. Such behavior is to be expected as the term $\mathcal{L}_{\text{Divide}}$ scatters the data and focuses on self-instance representation. When $\mathcal{L}_{\text{Rule}}$ is introduced, the model can restructure the embedding by linking similar instances. Also, we observe an augmentation in features, N_{feat}, that achieve statistical relevance for prognosis as we go through our learning procedure (for $K = 16$), which proves that our proposed framework can model more subtle patches interactions.

In Figure 3.8 we display the results over cohort $\mathcal{P}_A^{\text{clinical}}$ for the best performing model using HE reconstruction, $K = 16$ cluster, and all DNR losses. For each feature, we display the averaged hazard ratio (HR) using LOOCV. We solely depict the metrics that are selected by the CPH univariate fitting as statistically significant.

Out of the detected feature that contributes to the survival outcome of the patients, we observe different interactions between tissues. For example, we see blood vessels and tumor stroma ($\text{f}_{1,13}^{\text{inter}}$), which are linked to a lower survival outcome. A similar trend

is observed in the relation between tumor stroma and connective tissues ($\mathrm{f}_{2,8}^{\mathrm{inter}}$). Both predictions are linked to a deeper tumor invasion and, thus, lower survival outcomes. In the ablation studies, solely the $\mathrm{f}_{3}^{\mathrm{prob}}$ component is selected. The cluster shows the presence of tumor edges (*i.e.* outline of the tissue). However, such a feature should not be correlated to survival outcomes. After investigation, we observe that the feature is linked to the slide and tissue selection. Slides containing solely tumor are more likely to be sampled from more invasive cases and thus correlated with worst prognosis. As a result, the single-class distributions are not deterministic of survival.

3.5 Conclusion

In this chapter, we proposed a self-supervised learning method that offers a new approach to learning histopathological patterns within cancerous tissue regions. Our model presents a novel way to model the interactions between tumor-related image regions and tackles the inherent problem of data interpretability when predicting patient outcomes. Our method surpasses all previous baseline methods and achieves state of the art (SOTA) results in terms of C-Index without any data-specific annotation. Ablation studies also show the importance of different components of our method and the relevance of combining them.

However, we mention some limitations with our current approach. First, the proposed feature aggregation for patient description is based on a novel metric. Such features usually fail to be applied in daily diagnosis as experts do not trust them. Moreover, in practice, we often rely on data from multiple institutes or acquired with different scanners. This data heterogeneity causes a domain shift that can hinder the performance of downstream tasks. To reduce the domain shift, our presented architecture uses stain normalization to match feature distributions between the training and inference sets. This approach is time-consuming and prone to error as the normalization is usually computed region-based and thus inconsistent across the WSI. For instance, the normalization of dense tumor areas tends to fail due to their low expression of eosin.

In the next chapter, we tackle the problem of feature normalization and domain shift by combining unsupervised domain adaptation (UDA) and SSL.

4 Self-Rule to Multi Adapt: Handling Data from Multiple Sources

In the previous chapter, we explained how to take advantage of largely available unlabeled data to learn feature representation of colorectal cancer (CRC) tissues. However, we also mentioned the limitations of our approach when dealing with data from multiple sources. More specifically, let's assume we can access labeled data from an external source and want to transfer its knowledge to our inner target cohort. This situation commonly appears when dealing with publicly available data. The most straightforward approach would be to train a supervised model on the source-labeled data and apply it to our private data. However, as the public data comes from an external institute, the appearance of the whole slide images (WSIs) would most likely differ from the target site, which creates a domain shift. Unfortunately, this shift in distribution tends to lower the model's prediction quality.

One way to tackle the issue of domain shift is unsupervised domain adaptation (UDA). The approach works by learning from a rich source domain together with the label-free target domain to have a well-performing model on the target domain at inference time. UDA allows models to include a large variety of constraints to match relevant morphological features across the source and target domains.

Out of the recent works that rely on UDA, we cite the work of DANN [54] that uses gradient reversal layers to learn domain-invariant features. Self-Path [85] latter benefits from the DANN approach and combines it with self-supervised auxiliary tasks. The selected tasks reflect the structure of the tissue and are assumed to improve the stability of the framework when working with histopathological images. Such auxiliary tasks include hematoxylin channel prediction, jigsaw puzzle-solving, and magnification prediction. Another example is CycleGAN [164], which takes advantage of adversarial learning to map images between the source and target domain cyclically. However, adversarial approaches can fall short because they do not consider task-specific decision boundaries and only try to distinguish the features as either coming from the source or target domain [123]. A further issue is that most UDA methods consider fully-labeled source

datasets [48] for domain adaptation. However, digital pathology mainly relies on unlabeled or partly-labeled data, as acquiring fully labeled cohorts is often unfeasible. In addition, recent approaches treat domain adaptation as a closed-set scenario [21], which assumes that all target samples belong to classes present in the source domain, even though this is often not the case in a real-world scenario. To overcome this, OSDA [124] proposes an adversarial open-set domain adaptation approach, where the feature generator has the option to reject mistrusted or unknown target samples as an additional class. In another recent work, SSDA [153] uses self-supervised domain adaptation methods that combine auxiliary tasks, adversarial loss, and batch normalization calibration across the source and target domains.

In this chapter, we propose our label-efficient framework called Self-Rule to Multi Adapt (SRMA) [2] for tissue type recognition in histological images and attempt to overcome the issue of domain shift by combining self-supervised learning approaches with UDA (section 4.1). We present an entropy-based approach that progressively learns domain invariant features, thus making our model more robust to class definition inconsistencies as well as the presence of unseen tissue classes when performing domain adaptation. SRMA can accurately identify tissue types in hematoxylin and eosin (HE) stained images, which is an essential step for many downstream tasks. Our proposed method achieves this by using a few labeled open-source datasets and unlabeled data, which are abundant in digital pathology, thus reducing the annotation workload for pathologists. This work is an extension of our previously proposed Self-Rule to Adapt (SRA) [1] framework to multi-source domain adaptation (*i.e.* including an additional public dataset and performing further experiments to assess the model's performance). We show that our method outperforms previous domain adaptation approaches in a few-label setting and highlight the potential use for clinical application in the diagnostics of CRC in section 4.2. To promote open research, we make our code available on GitHub[1].

4.1 Method

In our unsupervised domain adaptation scenario, we have access to a small set of labeled data sampled from a source domain distribution and a set of unlabeled data from a target distribution. The goal is to learn a hypothesis function (*e.g.* a classifier) on the source domain that provides a good generalization in the target domain.

To this end, we propose a novel self-supervised cross-domain adaptation setting called SRMA, described in more detail below. We first introduce the architecture in a single-source setting in subsection 4.1.1 to subsection 4.1.4 and then present the generalization to the multi-source setting in subsection 4.1.5. Figure 4.1 gives an overview of the proposed framework, and algorithm 2 presents the pseudo-code of our SRMA method in the single-source setting.

[1] `https://github.com/christianabbet/SRA`.

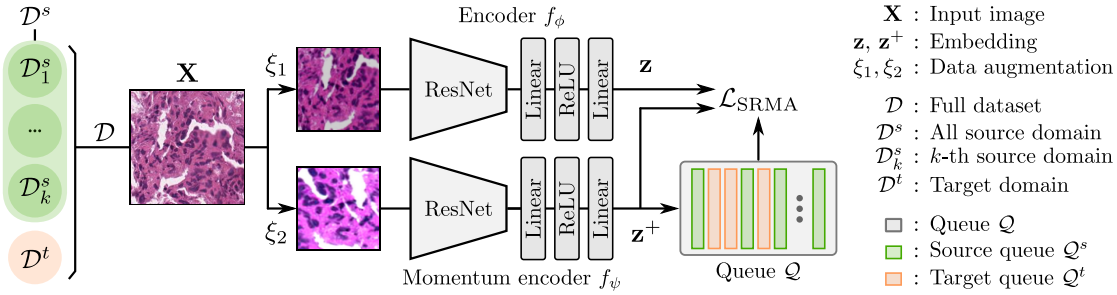


Figure 4.1 – Schematic overview of the Self-Rule to Multi Adapt (SRMA) framework for a given input image $\mathbf{X}$ sampled from $\mathcal{D} = \mathcal{D}_s \cup \mathcal{D}_t = \bigcup_{k=1}^{K} \mathcal{D}_s^k \cup \mathcal{D}_t$. Each encoder receives a different augmented version of the input image generated by transformations ξ_1 or ξ_2. The loss $\mathcal{L}_{\text{SRMA}} = \mathcal{L}_{\text{IND}} + \mathcal{L}_{\text{CRD}}$ is the composition of the in-domain loss ($\mathcal{L}_{\text{IND}}$) and cross-domain loss ($\mathcal{L}_{\text{CRD}}$), which aims at reducing the domain gap between the source and target domains. The queue $\mathcal{Q}$ keeps track of previous samples' embeddings and their set of origin (source or target).

4.1.1 Architecture

To train our framework, we rely on a set of images $\mathcal{D} = \mathcal{D}^s \cup \mathcal{D}^t$ that is the aggregation of a set of source images $\mathcal{D}^s$ and a set of target images $\mathcal{D}^t$. The model takes as input an RGB image $\mathbf{X} \in \mathbb{R}^{H \times W \times 3}$ sampled from $\mathcal{D}$ where H and W denote the height and width of the image, respectively. After sampling, two sets of random transformations are applied to the image $\mathbf{X}$ using image augmentations $\xi_1, \xi_2 : \mathbb{R}^{H \times W \times 3} \to \mathbb{R}^{H \times W \times 3}$. It generates a pair of augmented views that are assumed to share similar content as they are both different augmentations of the same sampled input image. Each image of the pair is then fed to its respective encoder $f_\phi : \mathbb{R}^{H \times W \times 3} \to \mathbb{R}^d$ and momentum encoder $f_\psi : \mathbb{R}^{H \times W \times 3} \to \mathbb{R}^d$ to compute the query $\mathbf{z} \in \mathbb{R}^d$ and key $\mathbf{z}^+ \in \mathbb{R}^d$ embeddings of the input image. Here, ϕ, ψ, and d represent the weights of the encoder, the weights of the momentum encoder, and the dimension of the embedding space, respectively. For notation simplicity, when sampling an image $\mathbf{X}$, we directly assume its embedding as $\mathbf{z}, \mathbf{z}^+ \in \mathcal{D}$.

Each network's branch consists of a residual encoder followed by two linear layers based on the architecture proposed in MoCoV2 [32]. We use the key embeddings $\mathbf{z}^+$ to maintain a queue of past samples $\mathcal{Q} = \{\mathbf{q}_l \in \mathbb{R}^d\}_{l=1}^{|Q|}$ in a first-in, first-out fashion, where Q is the size of the queue. The elements of $\mathcal{Q}$ are called *negatives* as they represent previously encoded entries that are different from the current batch elements. When updating the queue with a new negative sample, not only the sampled image's embedding is stored, but also its domain of origin (source or target). It allows the architecture to know at any time the domain of origin of each queue sample.

The queue provides many examples that alleviate the need for a large batch size [31] or the use of a memory bank [82]. In addition, it enables the model to scale more easily as

Algorithm 2: Pseudocode for the single-source SRMA framework.

Initialize queue $\mathcal{Q}$ by sampling from normal distribution ;
Normalize queue vectors $\{q_i\} \in Q$;
for e $= 0$ **to** $N_{\text{epochs}} - 1$ **do**
  Create $\mathcal{D}$ by uniformly sampling from $\mathcal{D}^s$ and $\mathcal{D}^t$;
  Update easy to hard coefficient r ; ▷ Equation 4.10
  for batch $\{\mathbf{X}_i \in \mathcal{D}\}_{i=1}^{B}$ **do**
    Get augmented samples using data augmentation ξ ;
    Perform forward pass using f_ϕ and f_ψ on augmented data to get $\mathbf{z}_i$ and $\mathbf{z}_i^+$ respectively ;
    Normalize vectors $\mathbf{z}_i$, $\mathbf{z}_i^+$;
    Compute in-domain loss $\mathcal{L}_{\text{IND}}$; ▷ Equation 4.4
    Calculate cross-entropy $\bar{H}$; ▷ Equation 4.7
    Compute easy to hard $\mathcal{R}^s$, $\mathcal{R}^t$ sets ; ▷ Equation 4.11
    Evaluate cross-domain loss $\mathcal{L}_{\text{CRD}}$; ▷ Equation 4.9 updated
    Compute $\mathcal{L}_{\text{SRA}} = \mathcal{L}_{\text{IND}} + \mathcal{L}_{\text{CRD}}$; ▷ Equation 4.1
    Update f_ϕ weights with backpropagation ;
    Update f_ψ weights with momentum ; ▷ Equation 4.17
    Update queue $\mathcal{Q}$ with $\mathbf{z}_i^+$; ▷ Equation 3.5
  end
end

$\mathcal{D}$ grows since the queue size does not depend on it. Moreover, f_ψ is updated using a momentum approach, combining its weights with those of f_ϕ. This approach ensures that f_ψ generates a slowly shifting and coherent embedding.

Motivated by recent work in the field [1, 55, 82], we extend the domain adaptation learning procedure to our model definition and task. Hence, we split the loss terms into two distinct tasks, namely the in-domain $\mathcal{L}_{\text{IND}}$ and cross-domain $\mathcal{L}_{\text{CRD}}$ representation learning. The objective loss $\mathcal{L}_{\text{SRMA}}$ is the summation of both terms, which are described in more detail below.

$$\min_{\phi} \mathcal{L}_{\text{SRMA}} = \min_{\phi} \mathcal{L}_{\text{IND}} + \mathcal{L}_{\text{CRD}}. \tag{4.1}$$

4.1.2 In-domain Loss

The first objective $\mathcal{L}_{\text{IND}}$ aims at learning the distribution of each source and the target domain features individually. We want to keep the two domains independent as their alignment is optimized separately by the cross-domain loss term. For each embedding vector $\mathbf{z}$, there is a paired embedding vector $\mathbf{z}^+$ generated from the same sampled tissue image and therefore is, by definition, similar. As a result, the sample's similarity can be jointly optimized using a contrastive learning approach [110]. Here, we strongly benefit from data augmentation to create discriminant features that match both $\mathbf{z}$ and

$\mathbf{z}^+$, making them more robust to outliers. By selecting data augmentations suited to histology [50, 136], we can ensure that the learned features are consistent with naturally occurring data variations in histology and, therefore, guide the model towards histopathologically meaningful representations. This approach differs from other recent works [82], where memory banks are used instead of the combination of a queue and data augmentation to keep track of past samples. The in-domain loss, as expressed in Equation 4.4, constrains the representation of the embedding space for each domain separately.

$$p^{\,\mathrm{IND}}(\mathbf{z}, \mathbf{z}^+, \mathcal{Q}) = \frac{\exp(\mathbf{z}^\top \mathbf{z}^+/\tau)}{\exp(\mathbf{z}^\top \mathbf{z}^+/\tau) + \sum\limits_{\mathbf{q}_l \in \mathcal{Q}} \exp(\mathbf{z}^\top \mathbf{q}_l/\tau)}. \tag{4.2}$$

$$l^{\,\mathrm{IND}}(\mathcal{D}, \mathcal{Q}) = \sum_{\mathbf{z}\,, \mathbf{z}^+ \in \mathcal{D}} \log \Big[p^{\,\mathrm{IND}}(\mathbf{z}\,, \mathbf{z}^+, \mathcal{Q}) \Big]. \tag{4.3}$$

$$\mathcal{L}_{\mathrm{IND}} = \frac{-1}{|\mathcal{D}_s| + |\mathcal{D}_t|} \left[l_{\,\mathrm{IND}}(\mathcal{D}_s, \mathcal{Q}_s) + l_{\,\mathrm{IND}}(\mathcal{D}_t, \mathcal{Q}_t) \right]. \tag{4.4}$$

We denote $\mathcal{Q}^s, \mathcal{Q}^t \subset \mathcal{Q}$ as the sets of indexed samples of the queue that were previously drawn from the corresponding domain $\mathcal{D}^s, \mathcal{D}^t \subset \mathcal{D}$, and $\tau \in \mathbb{R}$ as the temperature. The temperature is typically small ($\tau < 1$), thus sharpening the signal and helping the model to make confident predictions. For all images of each dataset $\mathcal{D}^s, \mathcal{D}^t$, we want to minimize the distance between $\mathbf{z}$ and $\mathbf{z}^+$ while maximizing the distance to the previously generated negative samples from the corresponding sets $\mathcal{Q}^s, \mathcal{Q}^t$. The samples in the queue are considered reliable negative candidates as they are generated by f_ψ whose weights are slowly optimized due to its momentum update procedure.

4.1.3 Cross-domain Loss

We can see the cross-domain matching task as the generation of features that are discriminative across both sets. In other words, two samples that are visually similar but are drawn from the source $\mathcal{D}^s$ and target $\mathcal{D}^t$ domain, respectively, should have a similar embedding. On the other hand, when comparing these samples to the remaining candidates of the opposite domain, their resulting embeddings should be far apart. Practically, performing cross-domain matching using the number of available candidates within a batch might deteriorate the quality of the domain-matching process due to the limited amount of negative samples. Therefore, we use the queue to find negative samples for domain matching. Hence, we compute the similarity and cross-entropy of each query pair $\mathbf{z}$, $\mathbf{z}^+$ drawn from one set (for example, $\mathcal{D}^s$) to the stored queue samples

from the other set (for example, $\mathcal{Q}^t$):

$$p^{\,\mathrm{CRD}}(\mathbf{z}, \mathbf{q}, \mathcal{Q}) = \frac{\exp(\mathbf{z}^\top \mathbf{q}/\tau)}{\sum\limits_{\mathbf{q}_l \in \mathcal{Q}} \exp(\mathbf{z}^\top \mathbf{q}_l/\tau)}, \tag{4.5}$$

$$H(\mathbf{z}, \mathbf{z}^+, \mathcal{Q}) = -\sum_{\mathbf{q} \in \mathcal{Q}} p^{\,\mathrm{CRD}}(\mathbf{z}, \mathbf{q}, \mathcal{Q}) \log \Big[p^{\,\mathrm{CRD}}(\mathbf{z}^+, \mathbf{q}, \mathcal{Q}) \Big], \tag{4.6}$$

$$\bar{H}(\mathbf{z}, \mathbf{z}^+, \mathcal{Q}) = \frac{1}{2} \Big[H(\mathbf{z}\,, \mathbf{z}^+, \mathcal{Q}) + H(\mathbf{z}^+, \mathbf{z}\,, \mathcal{Q}) \Big]. \tag{4.7}$$

A low cross-entropy H means that the selected query pair $\mathbf{z}\,, \mathbf{z}^+$ from one domain matches with a limited number of samples from another domain. The fact that the model matches the query with only a subset of samples of the other domain implies that it is confident when building domain-agnostic features to retrieve relevant candidates. Moreover, we update our initial definition of H in SRA [1], where solely $\mathbf{z}$ is used. By taking the average cross-entropy $\bar{H}$, the model is now also penalized when the predictions from $\mathbf{z}\,, \mathbf{z}^+$ of the same image are different. It improves the consistency of the domain matching [9]. As a result, the loss $\mathcal{L}_{\mathrm{CRD}}$ aims to minimize the averaged cross-entropy of the similarity distributions, assisting the model in making confident predictions:

$$l^{\,\mathrm{CRD}}(\mathcal{D}, \mathcal{Q}) = \sum_{\mathbf{z}\,, \mathbf{z}^+ \in \mathcal{D}} \bar{H}(\mathbf{z}, \mathbf{z}^+, \mathcal{Q}), \tag{4.8}$$

$$\mathcal{L}_{\mathrm{CRD}} = \frac{1}{|\mathcal{D}^s| + |\mathcal{D}^t|} \Big[l^{\,\mathrm{CRD}}(\mathcal{D}^s, \mathcal{Q}^t) + l^{\,\mathrm{CRD}}(\mathcal{D}^t, \mathcal{Q}^s) \Big]. \tag{4.9}$$

4.1.4 Easy-to-hard Learning

Two main pitfalls can hamper the performance of our cross-domain entropy minimization.

Firstly, at the start of the learning process, the similarity measure between samples and the queue is unclear as the model weights are initialized randomly, which does not guarantee proper feature descriptions. As a result, the optimization of their relative entropy and the loss term $\mathcal{L}_{\mathrm{CRD}}$ is ambiguous in the first epochs.

Secondly, being able to find matching samples for all input queries across datasets is a strong assumption. In clinical applications, we often rely on open-source datasets with a limited number of classes to annotate complex tissue databases. More specifically, challenging tissue types such as complex stroma subtypes are often absent in public

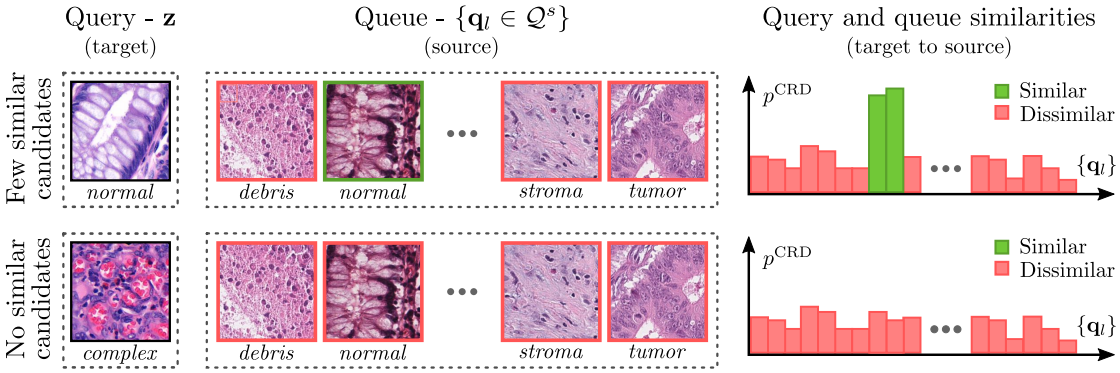


Figure 4.2 – Toy example of the cross-domain matching of different target queries to a fixed source queue. The first column shows two example target query images with computed embedding $\mathbf{z}$. The second column depicts the source queue images maintained by the model and their corresponding embeddings $\{\mathbf{q}_l\}$. In the third column, the distribution of the computed similarities p^{CRD} between the queries and each queue sample are plotted. Similar and dissimilar patterns with respect to the query are displayed in green and red. The top row highlights the case where the model is able to find at least a subset of elements of the queue that match the query (low entropy), as opposed to the bottom row where none of the queue samples match the presented query (high entropy). The class labels in this figure have been added for ease of reading and are unavailable during training.

datasets while frequent in the WSIs encountered in daily diagnostics. This example is illustrated in Figure 4.2. The top row shows the case where for a given target query $\mathbf{z}$ there are samples with a similar pattern in the source queue (*i.e.* the distribution of similarities p^{CRD} has low entropy). The second row highlights the opposite scenario where no queue elements match the query, generating a quasi-uniform distribution of similarities and, therefore, a high entropy. In other words, optimizing Equation 4.7 for all samples might result in a performance drop as we try to find cross-domain candidates even if there are none to be found.

We introduce an easy to hard (E2H) learning scheme to tackle both of these issues. The model starts with easy-to-match (low cross-entropy) samples and progressively includes harder (high cross-entropy) samples as the training progresses. We assume the model becomes more robust after each iteration and is more likely to properly process harder examples in later stages. Formally, we substitute the domains $\mathcal{D}^s, \mathcal{D}^t$ in Equation 4.9 with the corresponding set of candidates $\mathcal{R}^s, \mathcal{R}^t$ and update our cross-domain loss as:

$$r = \left\lfloor \frac{e}{N_{\text{epochs}} \cdot s_w} \right\rfloor \cdot s_h, \tag{4.10}$$

$$\begin{aligned}\mathcal{R}^s &= \{\mathbf{z}, \mathbf{z}^+ \in \mathcal{D}^s \mid \bar{H}(\mathbf{z}, \mathbf{z}^+, \mathcal{Q}^t) \text{ is reverse top-}r\},\\ \mathcal{R}^t &= \{\mathbf{z}, \mathbf{z}^+ \in \mathcal{D}^t \mid \bar{H}(\mathbf{z}, \mathbf{z}^+, \mathcal{Q}^s) \text{ is reverse top-}r\},\end{aligned} \tag{4.11}$$

$$\mathcal{L}_{\text{CRD}} = \frac{1}{|\mathcal{R}_s| + |\mathcal{R}_t|}\left[l^{\,\text{CRD}}(\mathcal{R}_s, \mathcal{R}_t) + l^{\,\text{CRD}}(\mathcal{R}_t, \mathcal{R}_s)\right]. \tag{4.9 updated}$$

where the ratio r is gradually increased during training using a step function. We denote s_w, s_h as the width and height of the step, respectively, N_{epochs} as the total number of epochs, and e the current epoch. The term reverse top-r indicates the ranking of cross-entropy terms in reverse order (low to high values). For example, $r = 0.2$ will capture the top 20% of the samples with the lowest cross-entropy. This definition ensures that as long as $r = 0$ (*i.e.* $N_{\text{epochs}} \cdot s_w > e$) we only use the in-domain loss $\mathcal{L}_{\text{IND}}$ for backpropagation, and the cross-domain loss term $\mathcal{L}_{\text{CRD}}$ is not considered. It lets us first learn feature representations based on the in-domain feature distribution. Moreover, with the tuning of the parameter s_h we can control the range of r and thus ensure that its value never reaches $r = 1$ to avoid systematic cross-domain matching where no candidates are available.

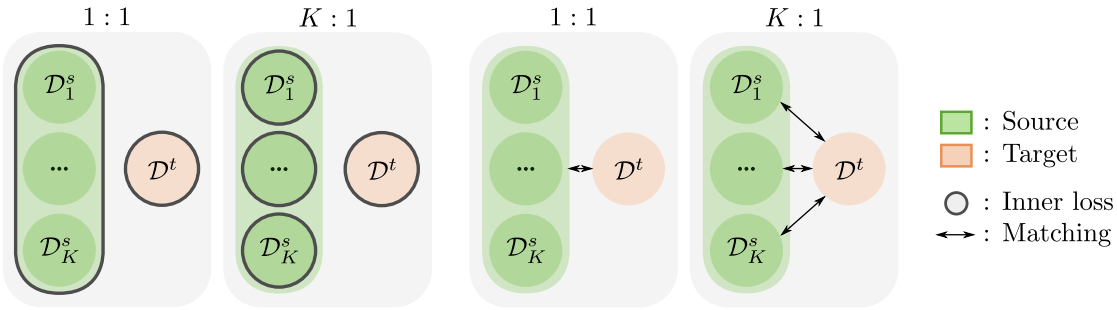
4.1.5 Generalization to Multiple Source Scenario

Our proposed SRMA framework can be generalized to consider multiple datasets in the source domain. It is especially useful if the available source datasets overlap in terms of class definitions, which increases the diversity of the visual appearance of the source data. More formally, we rely on K source datasets denoted $\mathcal{D}_k^s$ where $\bigcup_{k=1}^K \mathcal{D}_k^s = \mathcal{D}^s$, and $\mathcal{D} = \mathcal{D}^s \cup \mathcal{D}^t$. The same is valid for the source queues $\mathcal{Q}_k^s$ where $\bigcup_{k=1}^K \mathcal{Q}_k^s = \mathcal{Q}^s$, and $\mathcal{Q} = \mathcal{Q}^s \cup \mathcal{Q}^t$. We present two multi-source scenarios as depicted in Figure 4.3 for both the in-domain and cross-domain loss.

One option is to consider all source domains as a single domain $\mathcal{D}^s = \bigcup_{k=1}^K \mathcal{D}_k^s$ for the in-domain loss:

$$\mathcal{L}_{\text{IND}}^{1:1} = \frac{-1}{|\mathcal{D}^s| + |\mathcal{D}^t|}\left[l^{\,\text{IND}}(\bigcup_{k=1}^{K} \mathcal{D}_k^s, \bigcup_{k=1}^{K} \mathcal{Q}_k^s) + l^{\,\text{IND}}(\mathcal{D}^t, \mathcal{Q}^t)\right]. \tag{4.12}$$

Here, we make no distinction between the source sets and consider a one-to-one features representation importance $(1:1)$ between the source and target domain. This definition is equivalent to the single source in-domain adaptation. Alternatively, we can consider each source and the target domain as independent sets as in Equation 4.13. With this



(a) In-domain scenarios ($\mathcal{L}_{\text{IND}}$) (b) Cross-domain scenarios ($\mathcal{L}_{\text{CRD}}$)

Figure 4.3 – Proposed multi-source scenarios for (a) the in-domain $\mathcal{L}_{\text{IND}}$ and (b) the cross-domain $\mathcal{L}_{\text{CRD}}$ optimization. With the one-to-one settings $(1:1)$, we treat all source sets $\mathcal{D}_k^s$ as a single set $\mathcal{D}^s$. Each source domain is considered an independent set in the K-to-one $(K:1)$ setting.

K-to-one $(K:1)$ scenario, we have $K+1$ separate in-domain optimizations:

$$\mathcal{L}_{\text{IND}}^{K:1} = \frac{-1}{|\mathcal{D}^s| + |\mathcal{D}^t|} \left[\sum_{k=1}^{K} l^{\,\text{IND}}(\mathcal{D}_k^s, \mathcal{Q}^s) + l^{\,\text{IND}}(\mathcal{D}^t, \mathcal{Q}^t) \right]. \tag{4.13}$$

The same logic applies to the cross-domain matching. We can either consider a one-to-one correspondence between the unified source domain and the target domain as in Equation 4.14 or match each of the individual source domains to the target as in Equation 4.15.

$$\mathcal{L}_{\text{CRD}}^{1:1} = \frac{-1}{|\mathcal{D}^s| + |\mathcal{D}^t|} \left[l^{\,\text{CRD}}(\bigcup_{k=1}^{K} \mathcal{D}_k^s, \mathcal{Q}^t) + l^{\,\text{CRD}}(\mathcal{D}^t, \bigcup_{k=1}^{K} \mathcal{Q}_k^s) \right]. \tag{4.14}$$

$$\mathcal{L}_{\text{CRD}}^{K:1} = \frac{-1}{|\mathcal{D}^s| + K|\mathcal{D}^t|} \sum_{k=1}^{K} \left[l^{\,\text{CRD}}(\mathcal{D}_k^s, \mathcal{Q}^t) + l^{\,\text{CRD}}(\mathcal{D}^t, \mathcal{Q}_k^s) \right]. \tag{4.15}$$

The formulation of the E2H learning procedure has to be updated to comply with the multi-source domain definition. For the one-to-one setting, sets $\mathcal{R}^s$, $\mathcal{R}^t$ remain unchanged as we make no distinction between the different source sets. However, for the K-to-one setting, the model seeks to match the target domain to the source domain without considering multiple available source domains. We replace the domains $\mathcal{D}_k^s$, $\mathcal{D}^t$

in Equation 4.15 with the corresponding set of candidates $\mathcal{R}_k^s$, $\mathcal{R}^t$ defined as:

$$\begin{aligned} \mathcal{R}_k^s &= \{\mathbf{z}, \mathbf{z}^+ \in \mathcal{D}_k^s \mid \bar{H}(\mathbf{z}\,, \mathbf{z}^+, \mathcal{Q}^t) \text{ is reverse top-}r\}, \\ \mathcal{R}^t &= \bigcup_{k=1}^{K} \{\mathbf{z}, \mathbf{z}^+ \in \mathcal{D}^t \mid \bar{H}(\mathbf{z}\,, \mathbf{z}^+, \mathcal{Q}_k^s) \text{ is reverse top-}r\}. \end{aligned} \tag{4.16}$$

The overall loss $\mathcal{L}_{\text{SRMA}}$ for the multi-source setting is the combination of the in-domain loss ($\mathcal{L}_{\text{IND}}^{1:1}$ or $\mathcal{L}_{\text{IND}}^{K:1}$) and the cross-domain loss ($\mathcal{L}_{\text{CRD}}^{1:1}$ or $\mathcal{L}_{\text{CRD}}^{K:1}$).

4.2 Experiments

In this section, we present and discuss the experimental results. The general experimental setup is described in subsection 4.2.1. We validate our proposed self-supervised domain adaptation approach using publicly available datasets and compare it to current state of the art (SOTA) methods for UDA in subsection 4.2.2. Additionally, we assess the performance in a clinically relevant use case by validating our model on WSI sections from our in-house cohort in subsection 4.2.3. We perform an ablation study in subsection 4.2.4 for the single-source setting as well as additional experiments on the importance of the E2H learning procedure in subsection 4.2.5. These experiments are further extended to a multi-source application in subsection 4.2.6 to subsection 4.2.7 on both publicly available datasets and WSI sections.

4.2.1 Experimental Settings

In this section, we present the general setup that is used in all experiments. First, the architecture is trained in an unsupervised fashion and is referred to as the pretraining step. Next, a linear classifier is trained on top as described and is referred to as the classification step [31].

For the unsupervised learning step, the architectures of the feature extractors, f_ϕ and f_ψ, are composed of a ResNet18 [65] followed by two fully connected layers (projection head) using rectified linear unit (ReLU). The output dimension of the multi-layer projection head is $d = 128$. We update the weights of f_ϕ as θ_ϕ using standard backpropagation and f_ψ as θ_ψ with momentum $m = 0.999$:

$$\theta_\psi \leftarrow m\theta_\psi + (1 - m)\theta_\phi. \tag{4.17}$$

The model is trained from scratch for $N_{\text{epochs}} = 200$ epochs until convergence using the

stochastic gradient descent (SGD) optimizer (momentum = 0.9, weight decay = 10^{-4}), a learning rate of $\lambda = 0.03$, and a batch size of $B = 128$. The queue size is fixed to $Q = 2^{16} = 65,536$ samples. For the similarity learning, we set $\tau = 0.2$. We apply random cropping, grayscale transformation, horizontal/vertical flipping, rotation, grid distortion, ISO noise, Gaussian noise, and color jittering as data augmentations ξ_1, ξ_2. At each epoch, we sample $50,000$ examples with replacement from both the source and target dataset to create $\mathcal{D}$ with a total of $N = 100,000$ samples. The ratio r is updated between each epoch, while the sets $\mathcal{R}^s$, $\mathcal{R}^t$ for cross-domain matching are computed batch-wise.

During the second phase, the momentum encoder branch is discarded as it is not used for inference. The classification performance is evaluated using a linear classifier, which is placed on top of the frozen feature extractor. The linear classifier directly matches the output of the embedding d to the number of classes. It is trained for $N_{\text{epochs}} = 100$ epochs until convergence using the SGD optimizer (momentum = 0.9, weight decay = 0), a batch size of $B = 128$, and a learning rate of $\lambda = 1$. We use only a few randomly selected source labels to train this classification layer in order to simulate the clinical application, where we usually rely on a large quantity of unlabeled data and only have access to a few labeled samples. More precisely, we use $n = 1,000$ samples (*i.e.* 1%) to train the linear classifier with Kather 19 (K19) and $n = 500$ samples (*i.e.* 10%) when training with Kather 16 (K16). For the classes we use the following abbreviations: adipose (ADI), background (BACK), debris (DEB), lymphocytes (LYM), normal mucosa (NORM), mucin (MUC), muscle (MUS), stroma (STR), tumor (TUM), and complex stroma (CSTR). While training the linear classifier, we multi-run 10 times to obtain statistically significant results. The set of selected source labels varies between these runs, as they are randomly sampled for each run. If not specified otherwise, we use $s_w = 0.25$ and $s_h = 0.15$ for E2H learning. We use a ResNet18 backbone for all the presented baselines for a fair comparison.

The complex stroma class definitions between K16 and colorectal cancer tissue phenotype (CRCTP) are different. As a result, the complex stroma class is kept for training but excluded from the evaluation process when performing adaptation on K16 and CRCTP. With this problem definition, we fall into an open-set scenario where the class distribution of the two domains does not rigorously match, as opposed to a closed-set adaptation scheme.

In addition, we create an in-house cohort for domain adaptation that we name $\mathcal{D}_{\text{SRMA-WSI}}$. The dataset is composed of 665 HE-stained WSIs from our local CRC patient cohort $\mathcal{P}_A$. The slides originate from 383 unique patients diagnosed with adenocarcinoma and are scanned at a resolution of 0.248µm/px (40×). None of the selected slides originated from patients who underwent preoperative treatment. From each WSI, we uniformly sample 300 (448px × 448px, 111µm × 111µm) regions from the foreground masks to reduce the computational complexity of the proposed approach. It creates a dataset with $199,500$ unique and unlabeled patches. We assume that these randomly selected samples

Table 4.1 – Classification and unsupervised domain adaptation (UDA) from K19 (source) to K16 (target). The top results for the domain adaptation methods are highlighted in bold. We report the F_1 score for each class as well as the overall weighted F_1 score (W-F_1) averaged over 10 runs.

	Pretrain		Class.									
Methods	K19	K16	K19	K16	TUM	STR	LYM	DEB	NORM	ADI	BACK	W-F_1
Source only‡			✓		74.0**	77.4**	75.3**	50.5**	66.9**	87.0**	93.1**	75.1**
MoCoV2 [32]	✓		✓		93.5**	**79.3**$^{+}$	49.7**	**68.6**	91.6**	96.1**	96.0**	82.2**
MoCoV2 [32]†	✓	✓	✓		36.8**	45.4**	27.1**	30.8**	45.2**	43.1**	43.6**	38.9**
DANN [54]	✓	✓	✓		65.8**	60.8**	42.3**	47.8**	61.9**	64.1**	62.3**	57.8**
Stain norm. [96]	✓	✓	✓		77.8**	75.9**	68.2**	42.1**	75.1**	77.4**	87.6**	72.2**
CylceGAN [165]	✓	✓	✓		70.7**	71.6**	62.3**	47.6**	75.5**	89.0**	88.2**	72.4**
SelfPath [85]	✓	✓	✓		71.5**	68.8**	68.1**	57.6**	77.6**	82.3**	85.5**	73.1**
OSDA [124]	✓	✓	✓		82.0**	78.2*	83.6*	63.8**	80.3**	90.8**	93.2*	81.7**
SSDA - Rot [153]	✓	✓	✓		85.1**	78.5**	81.3**	**68.2**	88.7**	93.9**	96.5**	84.7**
SSDA - Jigsaw [153]	✓	✓	✓		90.0**	**81.2**	79.5**	64.4**	88.3**	94.2**	95.7*	84.9**
SENTRY [116]	✓	✓	✓		88.7**	74.4**	**86.0**	65.5*	91.5**	94.1**	**97.9**$^{+}$	85.7**
SRA [1]	✓	✓	✓		93.4**	72.9**	82.7*	**67.9**$^{+}$	96.5*	**97.0**$^{+}$	**97.2**$^{+}$	86.9*
SRMA	✓	✓	✓		**97.3**	**79.3**$^{+}$	80.2**	62.2**	**98.7**	**97.6**	**98.1**	**87.7**
Target only§				✓	94.6**	83.6**	92.6**	88.7**	95.4**	97.8^{+}	98.5^{+}	93.0**

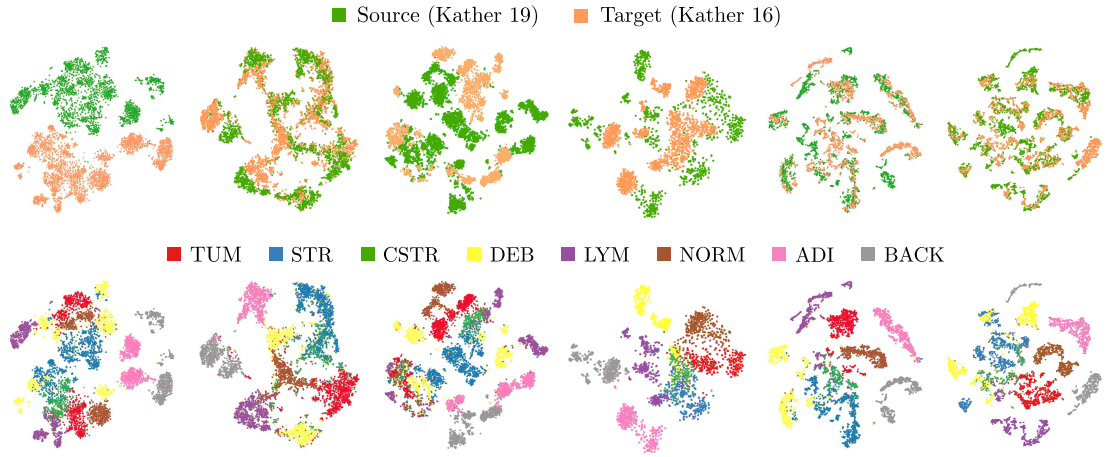
† Source and target datasets are merged and trained using contrastive learning.
‡ Direct transfer learning: trained on the source domain only, no adaptation (lower bound).
§ Fully supervised: trained knowing the labels of the target domain (upper bound).
$^{+}$ $p \geq 0.05$; * $p < 0.05$; ** $p < 0.001$; unpaired t-test with respect to the top result.

reasonably estimate our cohort's tissue complexity and heterogeneity.

4.2.2 Cross-domain Patch Classification

In this task, we use the larger dataset K19 as the source dataset and adapt it to K16. We motivate the selection of K19 as the source set by the fact that it is closer to the clinical scenario where we mainly rely on a large quantity of unlabeled data and only a few labeled ones, by using only 1% of the labels in K19. We evaluate the model's performance with the patch classification task on the K16 dataset. The mucin and muscle in K19 are grouped with debris and stroma, respectively, to allow comparison with the K16 class definitions. We use 70% of K16 to train the unsupervised domain adaptation. The remaining 30% are used to test the performance of the linear classifier trained on top of the self-supervised model.

The results of our proposed SRMA method are presented in Table 4.1, in comparison with baselines and SOTA algorithms for domain adaption. As the lower bound, we consider three approaches. Firstly, we apply direct transfer learning in a supervised fashion using the source data (source only). Secondly, we train MoCoV2 using the source domain as training data and apply it to the target domain. As the third baseline, we also use MoCoV2, but the model is trained on the source as well as the target domain,



(a) Source only (b) OSDA (c) SSDA (Jigsaw) (d) SENTRY (e) SRA (f) SRMA (ours)

Figure 4.4 – The t-SNE projection of the source (K19) and target (K16) domain embeddings. The top row shows the alignment between the source and target domain, while the bottom row highlights the representations of the different classes. We compare (f) our approach to other (b-e) UDA methods, and (a) the fully supervised, transfer learning baseline (source only).

merged into one training set. For the upper bound, we use the target domain data to train the model (fully supervised approach). The performances on complex stroma are not reported as the class is not present in K19. Figure 4.4 shows the t-SNE projection and alignment of the domain adaptation for the transfer learning (source only), the top-performing baselines (OSDA, SSDA with jigsaw solving), our previous work SRA and our novel approach (SRMA). Complementary results can be found in section C.1 and section C.2.

MoCoV2 fails to generalize knowledge between the sets when merging the source and target domains as it learns two distinct embeddings for each domain. The experiment highlights the limitations of contrastive learning without domain adaptation in the presence of domain distribution gaps. When training solely on the source domain, the contrastive approach shows better results and feature representations. Macenko stain normalization [96] slightly decreases the performances, compared to the source only baseline, as it introduces color artifacts that are very challenging for the network classifier. It mainly comes from the distribution of target samples, namely K16, composed of dark stained patches that are difficult to normalize properly.

CycleGAN suffers from performance degradation for the lymphocytes predictions. Like color normalization, it tends to create saturated images. In addition, the model alters the shape of the lymphocytes nuclei, thus fooling the classifier toward either debris or tumor classification.

In our setup, we observe that using the gradient reversal layer leads to an unstable loss

optimization for both Self-Path and DANN, which explains the large performance drops when training. Heavier data augmentations partially solve this issue. OSDA benefits from the open-set definition of the approach and achieves very good performance for lymphocytes detections. SSDA achieves similar results when using either rotation or jigsaw puzzle-solving as an auxiliary task. Due to the rotational invariance structure of the tissue and selected large magnification for tilling, rotation and jigsaw puzzle-solving are not optimal auxiliary tasks for digital pathology. Of the presented baselines, SENTRY achieves top competitive results on almost all classes. The main limitation appears to be the distinction between tumor and normal mucosa.

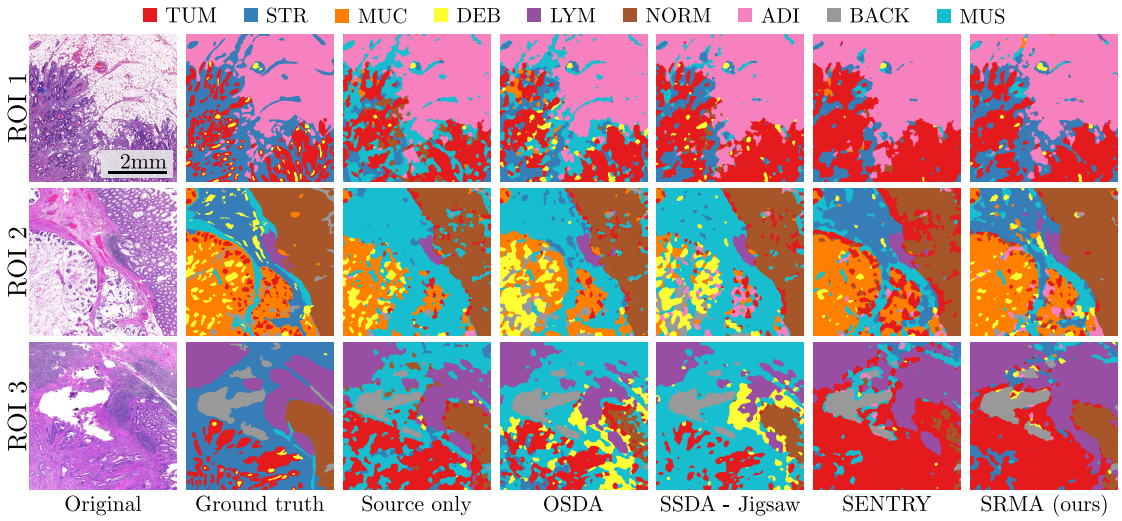
Our proposed SRMA method shows an excellent alignment between the same class clusters of the source and target distributions and outperforms SOTA approaches in terms of weighted F_1 score (W-F_1). Notably, our approach is even able to match the upper bound model for normal and tumor tissue identification. The embedding of complex stroma, which only exists in the target domain, is represented as a single cluster with no matching candidates, highlighting the model's ability to reject unmatchable samples from domain alignment.

Furthermore, the cluster representation is more compact compared to other presented methods, where, for example, normal mucosa tends to be aligned with complex stroma and tumor. Our approach suffers a drop in performance for stroma detection, which can be explained by the presence of lymphocytes in numerous stroma tissue examples, causing a higher misclassification rate. Moreover, the presence of loose tissue with a structure similar to stroma in the debris class is challenging. The overlap is also observed in the embedding projection.

4.2.3 Use Case: Cross-domain Segmentation of WSIs

In this section, we perform domain adaptation using our proposed model from K19 to our in-house dataset $\mathcal{D}_{\text{SRMA-WSI}}$. Moreover, we further validate our approach in a real case scenario on WSI ROIs. To do so, we select three ROIs of size 5mm $\times$ 5mm ($\simeq$ 20,000px $\times$ 20,000px), which an expert pathologist annotates according to the definitions used in the K19 dataset. The regions are selected such that, overall, they represent all tissue types, as well as challenging cases such as late cancer stage (ROI 1), mucinous carcinoma (ROI 2), and torn tissue (ROI 3). The annotated validation set is named $\mathcal{D}_{\text{SRMA-ROI}}$.

The qualitative and quantitative results are presented in Figure 4.5, alongside the original HE ROIs, their corresponding ground truth annotations, direct transfer learning (source only), as well as comparative results of the top-scoring SOTA approaches. We use a tile-based approach to predict classes on each ROI and use conditional random fields [25] to smooth the prediction map. The available labeled tissue regions are limited to the



(a) Qualitative results on region of interest (ROI).

	ROI 1			ROI 2			ROI 3		
Methods	Acc.	IoU	κ	Acc.	IoU	κ	Acc.	IoU	κ
Source only	58.2^{**}	51.5^{**}	0.450^{**}	53.0^{**}	42.9^{**}	0.436^{**}	42.9^{**}	34.9^{**}	0.366^{**}
OSDA [124]	62.7^{**}	56.8^{**}	0.503^{**}	47.9^{**}	37.5^{**}	0.389^{**}	37.2^{**}	33.3^{**}	0.317^{**}
SSDA - Jigsaw [153]	71.5^{**}	60.1^{**}	0.591^{**}	41.3^{**}	30.8^{**}	0.324^{**}	24.9^{**}	22.5^{**}	0.211^{**}
SENTRY [116]	68.4^{**}	53.6^{**}	0.520^{**}	63.7^{**}	52.5^{**}	0.551^{**}	47.0^{**}	33.5^{**}	0.379^{**}
SRMA (ours) [2]	**78.2**	**66.8**	**0.678**	**71.1**	**59.3**	**0.630**	**55.8**	**38.8**	**0.466**

$^{+}$ $p \geq 0.05$; * $p < 0.05$; ** $p < 0.001$; unpaired t-test with respect to top result.

(b) Quantitative results on region of interest (ROI).

Figure 4.5 – Qualitative (top) and quantitative (bottom) results of the domain adaptation from K19 to our unlabeled in-house dataset $\mathcal{D}_{\text{SRMA-WSI}}$. We show the original regions of interest (ROIs) from $\mathcal{D}_{\text{SRMA-ROI}}$ and their ground truth, respectively. We compare the performance of our SRMA algorithm to the lower bound and the top-performing SOTA methods. We report the pixel-wise accuracy, the weighted intersection over union (IOU), and the pixel-wise Cohen's kappa (κ) score averaged over 10 runs.

presented ROIs.

For all models, stroma and muscle are poorly differentiated as both have similar visual features without contextual information. This phenomenon is even more apparent in the source only setting, where muscle tissue is almost systematically interpreted as stroma. Moreover, due to the lack of domain adaptation, the boundary between tumor and normal tissues is not well defined, leading to incorrect predictions of these classes.

On the other hand, OSDA fails to adapt and generalize to new tumor examples while trying to reject mistrusted samples. This phenomenon is most visible in ROI 3, where the model interprets the surroundings of the cancerous region as a mixture of debris, stroma,

and muscle. SSDA tends to predict lymphocyte aggregates as debris. It can be explained by the model's sensitivity to staining variations as well as both classes' similarly dotted structures. Moreover, the model struggles to properly embed the representations of mucin. The scarcity of mucinous examples in the target domain makes the representation of this class difficult.

As in the patch classification task, SENTRY is the top-performing baseline. However, the model is still limited by its capacity to distinguish between tumor and normal mucosa due to the few label setting. Also, the detection of the stroma area appears less detailed compared to other approaches such as OSDA or SRMA.

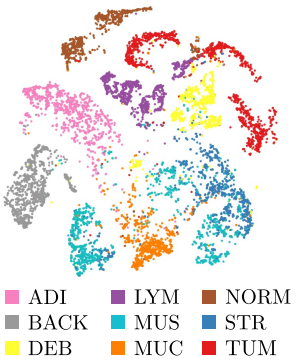
Our approach outperforms the other SOTA domain adaptation methods in terms of pixel-wise accuracy, weighed intersection over union (IOU), and pixel-wise Cohen's kappa score κ. Regions with mixed tissue types (*e.g.* lymphocytes/stroma or stroma/isolated tumor cells) are challenging cases because the samples available in the public cohorts mainly contain homogeneous tissue textures and few examples of class mixtures. Subsequently, domain adaptation methods naturally struggle to align features, resulting in a biased classification. We observe that thinner or torn stroma regions, where the background behind is well visible, are often misclassified as adipose tissue by SRMA, which is most likely due to their similar appearance. However, our SRMA model is able to correctly distinguish between normal mucosa and tumor, which are tissue regions with very relevant information downstream tasks such as survival analysis.

Figure 4.6 presents a qualitative visualization of the model's embedding space. The figure shows the actual visual distribution of the target patches, the source domain label arrangement, and the source and target domain overlap. The patch visualization also shows a smooth transition between class representations where, for example, neighboring samples of the debris cluster include a mixture of tissue and debris. The embedding reveals a large area in the center of the visualization that does not match the source domain. The area mostly includes loose connective tissue and stroma, which are both under-represented in the training examples. Also, mucin is improperly matched to the loose stroma, which explains the misclassification of stromal tissue in the ROI 2. The scarcity of mucinous examples in our $\mathcal{D}_{\text{SRMA-WSI}}$ cohort makes it difficult for the model to find suitable candidates.

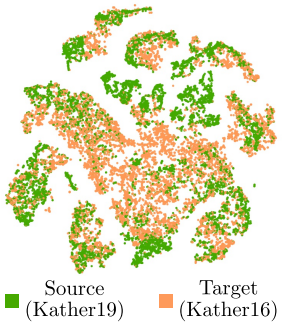
4.2.4 Ablation Study of the Proposed Loss Function

In this section, we present the ablation study of our SRMA approach. We denote $\mathcal{L}_{\text{IND}}$ as the in-domain loss, $\mathcal{L}_{\text{CRD}}$ as the cross-domain loss, and E2H as the easy-to-hard learning scheme. We evaluate the performance of our model on two tasks. The first one is the domain alignment between K19 (source) and K16 (target), which follows the experimental setting described in subsection 4.2.2. The results are presented in Table 4.2.

(a) t-SNE visualization of target patches



(b) t-SNE labeled source



(c) t-SNE source and target

Figure 4.6 – The t-SNE visualization of the SRMA model trained on K19 and our in-house data $\mathcal{D}_{\text{SRMA-WSI}}$. All sub-figures depict the same embedding. (a) Patch-based visualization of the embedding. (b) Distribution of the labeled source samples. (c) The relative alignment of the source and target domain samples.

The second task is the domain adaptation of K19 (source) to $\mathcal{D}_{\text{SRMA-WSI}}$ (target) and evaluation on $\mathcal{D}_{\text{SRMA-ROI}}$, as presented in subsection 4.2.3. Table 4.3 shows the results of these experiments. The following section jointly discusses the results of both tasks.

We use MoCoV2 [32] as a baseline. The model is trained following a contrastive learning approach just using the source domain data (K19) as well as with the source and target dataset merged.

We also compare our proposed approach SRMA to our previous work SRA [1]. For the single-source domain adaptation, the difference between SRA and the proposed extension SRMA lies in the reformulation of the cross-entropy matching. As a result, only the entropy-related terms, namely $\mathcal{L}_{\text{CRD}}$ and E2H, are affected. Thus, training SRA and SRMA using only the in-domain loss $\mathcal{L}_{\text{IND}}$ is the same set-up.

The baseline fails to learn discriminant features that match both sets, leading to poor performances in both cross-domain adaptation tasks. This shows that, if not constrained, the model is not able to generalize the knowledge and ends up learning two distinct

Table 4.2 – Ablation study for the proposed SRMA approach on classification. We denote $\mathcal{L}_{\text{IND}}$ as the in-domain loss, $\mathcal{L}_{\text{CRD}}$ as the cross-domain loss, and E2H as easy-to-hard. We train the domain adaptation from K19 (source) to K16 (target). We report the F_1 and weighted F_1 score (W-F_1) score for the individual classes and the overall mean performance (average over 10 runs).

Methods	$\mathcal{L}_{\text{IND}}$	$\mathcal{L}_{\text{CRD}}$	E2H	TUM	STR	LYM	DEB	NORM	ADI	BACK	W-F_1
MoCoV2 [32]†				93.5**	**79.3**$^{+}$	49.7**	**68.6**	91.6**	96.1**	96.0**	82.2**
MoCoV2 [32]‡				36.8**	45.4**	27.1**	30.8**	45.2**	43.1**	43.6**	38.9**
SRA [1]	✓			88.1**	72.8**	78.0*	71.8*	89.9**	93.4*	86.0*	82.9**
SRA [1]		✓		14.1**	9.1**	0.2**	10.1**	4.9**	0.0**	61.5**	14.4**
SRA [1]	✓	✓		63.0**	69.9**	**85.1**	57.7**	**98.2**$^{+}$	**97.9**	90.0**	80.3**
SRA [1]	✓	✓	✓	93.4**	72.9**	82.7*	**67.9**$^{+}$	96.5**	97.0**	97.2*	86.9*
SRMA		✓		35.3**	3.6**	0.0**	2.1	15.6**	64.0**	16.5**	19.8**
SRMA	✓	✓		93.3**	**77.4**$^{+}$	80.5**	**66.2**$^{+}$	91.4**	**97.8**$^{+}$	**98.3**	86.5*
SRMA	✓	✓	✓	**97.3**	**79.3**	80.2**	62.2**	**98.7**	**97.6**$^{+}$	**98.1**$^{+}$	**87.7**

† Trained on K19 only.
‡ K19 and K16 merged as a single set.
$^{+}$ $p \geq 0.05$; * $p < 0.05$; ** $p < 0.001$; unpaired t-test with respect to top result.

feature spaces, one for the source and one for the target domain.

Training using only $\mathcal{L}_{\text{IND}}$ achieves relatively good performances but fails to generalize knowledge to classes where textures and staining strongly vary. In the patch classification task, for example, this is apparent for the background and tumor class. For the second evaluation task, we can observe the same trend in the ROI 3 where the tumor and normal stroma are mixed.

Using only $\mathcal{L}_{\text{CRD}}$ does not help and creates an unstable model. As we do not impose domain representation, the model converges toward incorrect solutions where random sets of samples are matched between the source and target datasets. Moreover, it can create degenerated solutions where examples from the source and target domain are perfectly matched even though they do not present any visual similarity. The reformulation of the entropy, however, slightly improves the cross-domain matching.

Even the combination of the in-domain and cross-domain loss is not sufficient to improve the capability of the model. When performing a class-wise analysis, we observe that the performance on tumor and debris detection drastically dropped without the entropy reformulation. Both classes are forced to match samples from other classes, thus worsening the representation of the embedding.

The introduction of the E2H procedure improves the overall classification as well as most of the per-class performance for the first task. In the second task, it improves the performance across all metrics in all three ROIs. The importance of the E2H learning is

Table 4.3 – Ablation study for the proposed SRMA approach on WSI. We denote $\mathcal{L}_{\text{IND}}$ as the in-domain loss, $\mathcal{L}_{\text{CRD}}$ as the cross-domain loss, and E2H as easy-to-hard. We train the domain adaptation from K19 (source) to our in-house dataset (target). We report the pixel-wise accuracy, the weighted intersection over union (IOU), and the pixel-wise Cohen's kappa (κ) score for three manually annotated ROI (average over 10 runs).

				ROI 1			ROI 2			ROI 3		
Methods	$\mathcal{L}_{\text{IND}}$	$\mathcal{L}_{\text{CRD}}$	E2H	Acc.	IoU	κ	Acc.	IoU	κ	Acc.	IoU	κ
MoCoV2 [32]†	-	-	-	62.8^{**}	51.5^{**}	0.492^{**}	51.8^{**}	40.4^{**}	0.429^{**}	45.0^{**}	33.7^{**}	0.358^{**}
MoCoV2 [32]‡	-	-	-	55.6^{**}	47.0^{**}	0.417^{**}	29.8^{**}	19.8^{**}	0.220^{**}	32.1^{**}	25.5^{**}	0.240^{**}
SRA [1]§	✓	-	-	75.4^{*}	$\mathbf{65.5}^{+}$	0.646^{*}	67.9^{*}	55.1^{*}	0.594^{*}	49.8^{**}	35.7^{**}	0.415^{**}
SRA [1]§	-	✓	-	10.8^{**}	2.2^{**}	0.000^{**}	6.0^{**}	0.4^{**}	0.000^{**}	6.1^{**}	0.6^{**}	0.000^{**}
SRA [1]§	✓	✓	-	76.6^{*}	$\mathbf{66.0}^{+}$	0.658^{*}	70.1^{*}	58.2^{*}	0.619^{*}	52.6^{**}	36.8^{*}	0.438^{**}
SRA [1]§	✓	✓	✓	75.2^{*}	63.8^{*}	0.639^{*}	68.9^{**}	57.4^{**}	0.607^{**}	54.1^{*}	37.3^{*}	0.448^{*}
SRMA	-	✓	-	59.3^{**}	47.1^{**}	0.429^{**}	9.6^{**}	1.9^{**}	0.029^{**}	26.1^{**}	11.8^{**}	0.080^{**}
SRMA	✓	✓	-	72.4^{**}	63.4^{**}	0.608^{**}	$\mathbf{70.6}^{+}$	$\mathbf{59.1}^{+}$	$\mathbf{0.630}^{+}$	51.8^{**}	31.9^{**}	0.415^{**}
SRMA	✓	✓	✓	**78.2**	**66.8**	**0.678**	**71.1**	**59.3**	**0.630**	**55.8**	**38.8**	**0.466**

† Trained on K19 only.
‡ K19 and K16 merged as a single set.
$^{+}$ $p \geq 0.05$; * $p < 0.05$; ** $p < 0.001$; unpaired t-test with respect to top result.

evaluated and discussed in more detail in the next section.

Overall, the updated definition of the entropy improves the model's performance for both the cross-domain patch classification and WSI segmentation task. It helps to ensure that both model branches output a similar distribution, thus providing better cross-domain candidates. The improvement is most visible for the tumor and stroma predictions.

4.2.5 Evaluation of the E2H Learning Scheme

In this section, we discuss the usefulness and robustness of the E2H learning. The learning ratio r is based on the two contributing variables s_w and s_h. In Table 4.4, we show the impact of different combinations of these parameters on the single cross-domain segmentation task (see subsection 4.2.3). We report the pixel-wise accuracy, the weighted IOU, and the pixel-wise Cohen's kappa (κ) score for the presented ROI.

Firstly, we observe that the model is more robust when s_h is low. The variable is an indicator of the ratio of samples used for cross-domain matching. In other words, the architecture benefits from a small s_h that allows it to focus on examples with high similarity/confidence while avoiding complex samples without properly matching candidates. Secondly, the selection of s_w is also crucial to the stability of the prediction. This quantity measures the number of epochs to wait before considering more complex examples in the cross-domain matching optimization. For small s_w values, the model has

Table 4.4 – Importance of s_w and s_h parameter tuning for the E2H learning scheme on the three ROI for each parameter pair. We report the pixel-wise accuracy, the weighted intersection over union (IOU), and the pixel-wise Cohen's kappa (κ) score (average over 10 runs).

	Metrics								
Images	Acc.	IOU	κ	Acc.	IOU	κ	Acc.	IOU	κ
	$s_w = 0.125\,, s_h = 0.075$			$s_w = 0.125\,, s_h = 0.1$			$s_w = 0.125\,, s_h = 0.125$		
ROI 1	**0.777**$^{+}$	0.658*	**0.670**$^{+}$	0.758*	0.642**	0.646**	0.752**	0.652*	0.643**
ROI 2	0.686**	0.567**	0.602**	0.653**	0.527**	0.561**	0.697*	0.565*	0.613*
ROI 3	0.544*	0.375*	0.452*	0.542*	**0.388**$^{+}$	**0.458**$^{+}$	0.546*	0.369*	0.454*
ALL	0.669*	0.518**	0.618**	0.651**	0.495**	0.599**	0.665**	0.509**	0.615**
	$s_w = 0.25\,, s_h = 0.15$			$s_w = 0.25\,, s_h = 0.2$			$s_w = 0.25\,, s_h = 0.25$		
ROI 1	**0.782**$^{+}$	**0.668**$^{+}$	**0.678**$^{+}$	0.764*	0.642**	0.654*	0.756*	0.633**	0.642**
ROI 2	**0.711**$^{+}$	**0.593**	**0.630**$^{+}$	**0.709**$^{+}$	0.581*	**0.626**$^{+}$	0.703*	0.573*	**0.620**$^{+}$
ROI 3	**0.558**	**0.388**	**0.466**	**0.552**$^{+}$	**0.379**$^{+}$	**0.464**$^{+}$	0.542*	**0.384**$^{+}$	**0.459**$^{+}$
ALL	**0.684**	**0.535**$^{+}$	**0.635**	0.675*	0.521**	0.626**	0.667**	0.511**	0.617**
	$s_w = 0.5\,, s_h = 0.45$			$s_w = 0.5\,, s_h = 0.6$			$s_w = 0.5\,, s_h = 0.75$		
ROI 1	**0.786**	**0.680**	**0.684**	0.758**	0.641**	0.646**	0.745**	0.626**	0.629**
ROI 2	**0.714**	**0.589**$^{+}$	**0.631**	0.697*	0.563**	0.610*	0.697*	0.571*	0.614*
ROI 3	0.534**	**0.380**$^{+}$	0.447*	0.524**	0.370*	0.439**	0.520**	0.364*	0.438**
ALL	**0.678**$^{+}$	**0.539**	**0.629**$^{+}$	0.659**	0.510**	0.609**	0.654**	0.496**	0.603**

$^{+}$ $p \geq 0.05$; * $p < 0.05$; ** $p < 0.001$; unpaired t-test with respect to top result.

no time to learn the feature representation properly before encountering more difficult samples. This is especially true for the first few epochs after initialization, where the architecture is not yet able to optimally embed features. Furthermore, using large s_w weakens the model's capability to progressively learn from more complex samples.

Figure 4.7 shows an example patch from the training phase and highlights the usefulness of the E2H scheme. When dealing with a heterogeneous target data cohort, some tissue types might not have relevant candidates in the other set (open-set scenario). The presented example shows an example composed of a vein and blood cells. Such a tissue structure is absent from the source cohort and thus does not have a matching sample in the target domain.

Without the E2H learning, the model is forced to find matching candidates for the query $\mathbf{z}$, here normal mucosa (NORM), to minimize the cross-entropy term $\bar{H}$. When plotting the similarity distribution, the matched samples form an out-of-distribution cluster with high similarity to the query ($\mathbf{z}^{\top}\mathbf{q}_l \simeq 1$). This phenomenon is even more visible with the cumulative function (red) that tends to the step function.

When training with the E2H scheme, we observe a continuous transition in the distribution of sample similarities. Here, the top retrieved samples share the same granular structure

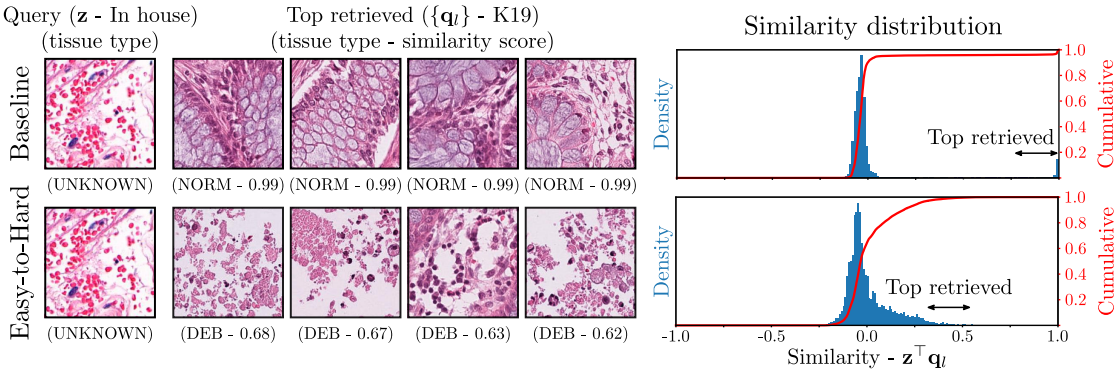


Figure 4.7 – Importance of the E2H learning scheme for the cross-domain image retrieval. The first column shows the input query image $\mathbf{z}$ from $\mathcal{D}_{\text{SRMA-WSI}}$ (target domain), the second column presents the retrieved samples from K19 that have the highest similarity in the source queue $\{\mathbf{q}_i\}$, and the third column shows the density distribution (blue) of similarities across the source queue as well as its cumulative profile (red). We list the retrieved examples with their assigned classes. The query class is unknown.

as the query. Still, we have to be careful as they do not represent the same type of tissue. The retrieved samples are examples of necrosis, whereas the query shows red blood cells. The fact that the architecture is less confident (i.e., the similarity is lower for the top retrieved samples) is a good indicator of its robustness and ability to process complex queries.

As a result, the introduction of the E2H process prevents the model from learning degenerated solutions. We also observe this with other open-set tissue classes, such as complex stroma and loose connective tissue, which are absent in the source domain.

4.2.6 Multi-source Patch Classification

We explore the benefit of using multiple source domains with different distributions to perform domain adaptation for the patch classification task. To do so, we select K19 and K16 as the source sets and CRCTP as the target set. To learn the feature representations, the model is trained in an unsupervised fashion using both source domains as well as the unlabeled target domain. For the evaluation, we train a linear classifier on top of the frozen features with a few randomly selected labeled samples from the source domains (1000 samples from K19 (1%), and 500 samples from K16 (10%)). By using only little labeled data, we aim to reduce the annotation workload for pathologists while still achieving good classification performances. The set of labeled data differs between each run, as they are randomly sampled for each individual run.

The three datasets K19, K16, and CRCTP do not have one-to-one classes correspondence. Thus, for the evaluation of the target set, we only consider the classes present in all datasets, namely, tumor (TUM), stroma (STR), lymphocytes (LYM), normal mucosa

Table 4.5 – Performance of the SRMA framework on the CRCTP dataset in a multi-source domain setting. We show the results for different combinations of K16 and K19 used for the self-supervised pre-training as well as training the classification header. We also compare the performance of the $1:1$ with the $K:1$ setting for the loss definitions (see Equations 4.12-4.15). We report the F1 score for the individual classes and weighted F_1 score (W-F_1) for the overall mean performance (all) (averaged over 10 runs).

	Pretrai		Class.		Multi-source							
Methods	K19	K16	K19	K16	$\mathcal{L}_{\text{IND}}$	$\mathcal{L}_{\text{CRD}}$	TUM	STR†	LYM	NORM	DEB†	W-F_1
Single source:												
SRA [1]	-	✓	-	✓	-	-	**82.2**	**69.3**	62.5**	**69.8**	47.4*	**69.4**
SRMA	-	✓	-	✓	-	-	**82.0**$^{+}$	63.5**	**66.3**	51.9**	**50.3**	65.2**
SRA [1]	✓	-	✓	-	-	-	91.0*	84.9**	62.0**	**71.7**	**58.5**$^{+}$	79.2**
SRMA	✓	-	✓	-	-	-	**91.7**	**86.7**	**65.4**	68.6**	**58.9**	**80.2**
Multi source:												
DeepAll [48]	✓	✓	-	✓	-	-	52.4**	64.1**	36.5**	14.2**	13.8**	47.1**
SRA [1]	✓	✓	-	✓	$1:1$	$1:1$	70.9**	68.5**	45.6**	72.2**	19.1**	62.2**
SRMA	✓	✓	-	✓	$1:1$	$1:1$	76.6**	69.3**	48.7**	74.5**	18.2**	64.4**
SRMA	✓	✓	-	✓	$K:1$	$1:1$	**89.4**$^{+}$	**74.9**$^{+}$	**66.8**	**75.6**	**43.7**	**74.4**
SRMA	✓	✓	-	✓	$1:1$	$K:1$	75.9**	73.3*	45.9**	73.0**	22.6**	65.8**
SRMA	✓	✓	-	✓	$K:1$	$K:1$	**89.8**	**75.2**	64.5**	74.1**	25.7**	72.5**
DeepAll [48]	✓	✓	✓	-	-	-	72.4**	88.6**	43.6**	53.2**	71.8**	73.2**
SRA [1]	✓	✓	✓	-	$1:1$	$1:1$	86.2**	87.6**	66.7**	71.0**	**80.5**	81.8**
SRMA	✓	✓	✓	-	$1:1$	$1:1$	**92.5**	88.4**	68.7**	68.3**	74.2*	82.9*
SRMA	✓	✓	✓	-	$K:1$	$1:1$	91.5*	87.6**	**70.7**	**75.0**	65.7**	82.7*
SRMA	✓	✓	✓	-	$1:1$	$K:1$	90.1**	**90.1**	**69.6**$^{+}$	72.9**	71.6**	**83.6**
SRMA	✓	✓	✓	-	$K:1$	$K:1$	**91.6**$^{+}$	87.4**	68.7**	73.9**	53.3**	81.2**
DeepAll [48]	✓	✓	✓	✓	-	-	81.4**	**85.7**$^{+}$	50.9**	50.1**	51.5**	72.6**
SRA [1]	✓	✓	✓	✓	$1:1$	$1:1$	85.8**	**85.9**	72.9*	72.1**	**59.2**	**80.1**
SRMA	✓	✓	✓	✓	$1:1$	$1:1$	**92.9**	82.4**	72.1*	70.8**	53.7**	79.3*
SRMA	✓	✓	✓	✓	$K:1$	$1:1$	**92.8**$^{+}$	81.7**	**73.5**	**74.6**	49.8**	79.3*
SRMA	✓	✓	✓	✓	$1:1$	$K:1$	89.6**	84.7*	72.5*	**74.4**$^{+}$	52.1**	**80.0**$^{+}$
SRMA	✓	✓	✓	✓	$K:1$	$K:1$	92.5*	80.6**	70.5**	73.9**	39.4**	77.4**

† The STR and MUS classes are merged as STR class; DEB and MUC classes as DEB.
$^{+}$ $p \geq 0.05$; * $p < 0.05$; ** $p < 0.001$; unpaired t-test with respect to top result.

(NORM), and debris (DEB). Still, during the unsupervised pre-training, we consider all classes, including those that do not have matching candidates across the sets, such as background (BACK) and adipose (ADI). This setup creates an open-set scenario for cross-domain matching and allows the model to learn more robust feature representations.

For comparison purposes, we use the same hyper-parameters as in the single source domain patch classification setting with $s_w = 0.25$, $s_h = 0.15$. The probability of drawing a sample $\mathbf{X}$ from the source or the target domain is the same. The results are

presented in Table 4.5. We compare the performance of different experimental setups in regards to the used datasets and multi-source scenario for our SRMA. We show three scenarios where we use either K16, K19, or the combination of the two (K16 and K19) to train the classification layer. To evaluate the impact of the multi-source scenario, where we investigate all possibilities for the in-domain ($\mathcal{L}_{\text{IND}}^{1:1}, \mathcal{L}_{\text{IND}}^{K:1}$) and cross-domain ($\mathcal{L}_{\text{CRD}}^{1:1}, \mathcal{L}_{\text{CRD}}^{K:1}$) loss definitions, as introduced in Equation 4.12 to Equation 4.15. As baselines, we consider the single source setting of the presented SRMA model, our previous SRA work, as well as the DeepAll approach that uses aggregation of all the source tissue data into a single training set [48].

The SRMA and SRA single source baselines both show a better performance for K19 compared to K16. It is most likely due to the fact that the variety of examples in K16 is limited (only $5,000$ examples), thus hindering the generalization of feature representations in the pre-training stage. Also, SRMA outperforms our previous SRA work for all classes except one, which indicates the entropy reformulation's robustness.

For the multi-source adaptation, we show three scenarios where we use either K16, K19, or the combination of the two (K16 and K19) to train the classification layer. When using solely K16, we can observe that the debris classification tends to have lower performances across all models. Debris examples in K16 appear highly saturated, making class generalization challenging. Only the proposed SRMA approach is able to achieve better performances compared to the single source baselines. Using K19 for the classification of target patches gives overall the best performance. Interestingly, using both K19 and K16 leads to a drop in performance. It is likely due to potential discrepancies between the class definitions, which makes it more difficult for the model to generalize the class representations across the different modalities.

When comparing the in-domain and cross-domain multi-source scenarios, we find that using $\mathcal{L}_{\text{IND}}^{1:1}$ and $\mathcal{L}_{\text{CRD}}^{K:1}$ achieves the best results across the various settings. It suggests optimizing the source domain as a single set for the in-domain representation is better. However, when performing cross-domain matching, considering domain-to-domain correspondence between each source set and the target domain yields better performances. It ensures that the model looks for relevant candidates in all individual source sets, as tissue samples might have a distinct appearance in different source domains.

We also note that $\mathcal{L}_{\text{IND}}^{K:1}$ is only relevant when only using K16 to train the classification header. It is because the cross-domain matching fails to retrieve debris samples correctly from the K16 domain, which tend to be misclassified as lymphocytes because of their similar granular appearance and as well as their hematoxylin-positive aspect. Overall the combination of both $\mathcal{L}_{\text{IND}}^{K:1}$ and $\mathcal{L}_{\text{CRD}}^{K:1}$ degrades the performance slightly. Complementary results on the importance of the dataset ratios when sampling data for the unsupervised pre-training phase are available in section C.3.

4.2.7 Use Case: Multi-source Segmentation of WSIs

In this section, we present the results for the multi-source domain adaptation for patch-based segmentation of WSI ROIs. More specifically, we are interested in the detection of desmoplastic reactions (complex stroma), which is a prognostic factor in CRC [138]. We use both K19 and CRCTP as the source datasets to add complex stroma examples to the source domain. Our in-house dataset $\mathcal{D}_{\text{SRMA-WSI}}$ is used as the target domain.

To assess the quality of the prediction, we evaluate the models on the same ROIs as in the single-source setting. However, the previously provided annotations do not include complex stroma. We overcome this by defining a margin around the tumor tissue in the existing annotations, which is considered as the interaction area. Stroma in this region is, therefore, re-annotated as complex stroma. The margin is fixed to 500µm such that it includes the close tumor neighborhood [13, 106]. Note that this is a rough estimation as the tumor-to-stroma interaction areas might vary a lot depending on the type of tumor.

As a baseline, we use DeepAll, which aggregates all the source tissue data into a single training set [48]. The model is trained in an unsupervised fashion using a standard contrastive loss to optimize the data representation of the features [32]. In this case, no domain adaption is performed across the sets.

The results are presented in Table 4.6 and Figure 4.8. In Table 4.6, we compare the performance of the models with and without complex stroma detection across all three ROIs. We compare the single as well as the multi-source SRMA approaches to the baselines, DeepAll, and our previously published SRA method. We report the F1-score for complex stroma, the overall weighted F_1 score (W-F_1), the pixel-wise accuracy, the Dice (DSC), the weighted IOU, and pixel-wise Cohen's kappa (κ).

Without considering the complex stroma class, the numerical results show that all the multi-source settings achieve similar performances. Including an additional dataset, namely CRCTP, does not improve nor seriously deteriorate the classification performances on the ROIs. Furthermore, merging the source domains for in-domain optimization ($\mathcal{L}_{\text{IND}}^{1:1}$) seems to be the best setup. For the cross-domain matching, both $\mathcal{L}_{\text{CRD}}^{1:1}$ and $\mathcal{L}_{\text{CRD}}^{K:1}$ achieve similar scores.

However, the benefit of using the multi-source approach can be observed when including complex stroma detection. Here, the models that use CRCTP as source set achieve better results. The detection of complex stroma improves by up to $20 - 25\%$. By contrast, the cross-domain matching on each subset $\mathcal{L}_{\text{CRD}}^{K:1}$ penalizes the complex stroma detection. It can be explained by the fact that only CRCTP contains examples of complex stroma. Therefore, imposing complex stroma retrieval in K19 is unfeasible. Another challenge is the relatively significant overlap between the complex stroma and the tumor class. The model tends to classify the tumor border area as complex stroma.

Table 4.6 – Analysis of the performance of the SRMA approach in regards to complex stroma detection. Multiple possible scenarios are evaluated in regard to the data included for pre-training, as well as the multi-source setting ($1:1$ versus $K:1$, see Equation 4.12 to Equation 4.15), as indicated in the table. We report the F1-score for complex stroma, the overall weighted F_1 score (W-F_1), the pixel-wise accuracy, the Dice (DSC) score, the intersection over union (IOU), and the pixel-wise Cohen's kappa (κ) score (averaged over 10 runs).

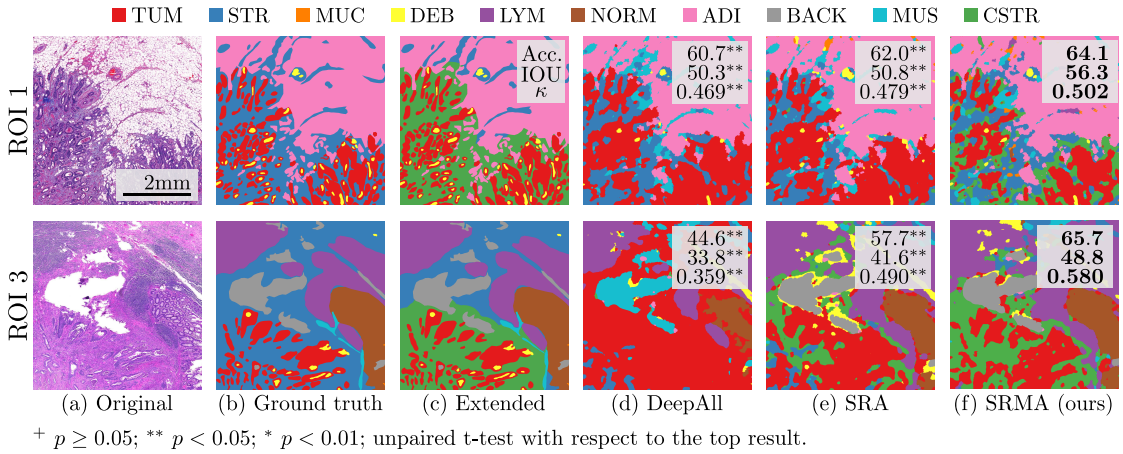
	Pretraining		Multi-source							
Model	K19	CRCTP	$\mathcal{L}_{\text{IND}}$	$\mathcal{L}_{\text{CRD}}$	F1-CSTR†	W-F_1	Acc.	DSC	IOU	κ
ROI 1-3 (w/o CSTR)										
DeepAll [48]	✓	✓	-	-	-	62.2**	61.5**	58.3**	48.3**	0.552**
SRA [1]	✓	-	-	-	-	64.8**	66.1**	63.2**	52.1**	0.611**
SRMA	✓	-	-	-	-	**66.7**$^{+}$	**68.4**$^{+}$	64.7**	**53.6**$^{+}$	**0.636**$^{+}$
SRMA	✓	✓	$1:1$	$1:1$	-	**67.3**	**68.5**	**66.9**	**54.1**	**0.636**
SRMA	✓	✓	$K:1$	$1:1$	-	64.4**	66.5**	63.7**	51.6**	0.615**
SRMA	✓	✓	$1:1$	$K:1$	-	**66.2**$^{+}$	**67.8**$^{+}$	65.2*	52.8*	**0.629**$^{+}$
SRMA	✓	✓	$K:1$	$K:1$	-	63.8**	66.0**	63.2**	50.9**	0.609**
ROI 1-3 (w/ CSTR)										
DeepAll [48]	✓	✓	-	-	0.1**	50.5**	53.9**	49.6**	39.9**	0.479**
SRA [1]	✓	-	-	-	21.4**	60.0**	62.4**	58.2**	49.0**	0.577**
SRMA	✓	-	-	-	26.3**	61.4**	64.1**	59.5**	49.8**	59.4**
SRMA	✓	✓	$1:1$	$1:1$	**47.9**$^{+}$	**64.7**$^{+}$	65.9*	**63.1**	**52.4**$^{+}$	0.613**
SRMA	✓	✓	$K:1$	$1:1$	**49.2**	**65.0**	**66.9**	61.8**	**52.4**	**0.624**
SRMA	✓	✓	$1:1$	$K:1$	**46.4**$^{+}$	**64.0**$^{+}$	65.1**	61.9*	51.3*	0.604**
SRMA	✓	✓	$K:1$	$K:1$	36.6**	62.3**	64.6**	59.7**	50.0**	59.9**

† Performances are only available with extended annotations (w/CSTR).
$^{+}$ $p \geq 0.05$; * $p < 0.05$; ** $p < 0.001$; unpaired t-test with respect to top

In Figure 4.8, we display the visual results of the complex stroma detection on ROIs 1 and 3, where desmoplastic reactions, and thus complex stroma, are present. We show, from left to right, the reference images, the original ground truth labels, the extended ground truth labels with complex stroma, the DeepAll baseline, our previous SRA work, and as well the results of the presented SRMA model ($\mathcal{L}_{\text{IND}}^{1:1}$ and $\mathcal{L}_{\text{CRD}}^{K:1}$ setting).

SRMA outperforms the baselines in terms of pixel-wise accuracy, IOU, and Cohen's kappa score κ. Notably, the detection of the tumor is much more detailed compared to the single-source approach in both ROIs. Parts of the tissue previously considered as tumors can now be properly matched, thanks to the introduction of the complex stroma class.

Another interesting result in ROI 3 is that all the stromal areas are now considered as either complex stroma, tumor, or lymphocytes by all models. It highlights how challenging the classification of complex stroma is without access to the higher-level



(a) Original (b) Ground truth (c) Extended (d) DeepAll (e) SRA (f) SRMA (ours)

$^{+}$ $p \geq 0.05$; ** $p < 0.05$; * $p < 0.01$; unpaired t-test with respect to the top result.

Figure 4.8 – Results of the multi-source domain adaptation from K19 and CRCTP to $\mathcal{D}_{\text{SRMA-WSI}}$ dataset. (a-c) show the original ROIs from $\mathcal{D}_{\text{SRMA-ROI}}$, their original ground truth (without CSTR), and the extended ground truth (with CSTR), respectively. We compare the performance of our SRMA framework (f) to our previous work SRA (e) and to the DeepAll baseline (d). For the multi-source optimization, we use the $1:1$ and $K:1$ approach for the in-domain and cross-domain, respectively. We report the pixel-wise accuracy, the weighted IOU, and the pixel-wise Cohen's kappa (κ) score averaged over 10 runs.

context. Pathologists also find this difficult, as they rely not only on the tissue morphology for this assessment but also on the spatial relations (*i.e.* the proximity to the tumor area). Here, according to our extended ground truth, the complex stroma only surrounds the tumor region. However, the tissue tear disconnected some of the tumor's surrounding regions, which suggests that the complex stroma area, in reality, spans even further. This correlates with the prediction of both models, which identify the whole region as complex stroma.

Lastly, using the multi-source setting allows the introduction of a new class as complex stroma to the detection task. In the presented setting, the source domains do not need one-to-one class correspondences for the model to learn meaningful cross-domain features. Here, CRCTP does not include mucin, background, and adipose, while K19 does not contain complex stroma. It is an interesting outcome, as it shows that new data that might even be acquired under different circumstances can be added with additional tissue classes without interfering with or altering the performance of the existing classes.

The visualizations of the multi-source domain embedding space as well as the patch-based segmentation of a full WSI image are available in section C.4-section C.5.

4.3 Conclusion

In this work, we explore the usefulness of self-supervised learning and UDA for the identification of histological tissue types. Motivated by the difficulty of obtaining expert annotations, we explore different UDA models using a variety of label-scarce colorectal cancer histopathology datasets from publicly available sources.

As our main contribution, we present a new label-transferring approach from partially labeled public datasets (source domain) to unlabeled target domains. It is more practical than most previous UDA approaches, which are often tailored to fully annotated source domain data or tied to additional network branches dedicated to auxiliary tasks. Instead, we perform progressive cross-entropy minimization based on the similarity distribution among the unlabeled target and source domain samples, yielding discriminative and domain-agnostic features for domain adaptation.

Throughout various label transfer tasks, we show that our proposed SRMA method can discover the relevant semantic information even in the presence of few labeled source samples and yields a better generalization on different target domain datasets. Moreover, we show that our model definition can be generalized to a multi-source setting. As a result, the proposed model is able to learn rich data representation using multiple source domains.

So far, the presented datasets are mainly composed of curated and, thus, homogeneous tissue. Such data, however, does not capture the heterogeneity and complexity of patches extracted from images in the diagnostic routine. It can lead to erroneous detection, *e.g.*, background, and stroma interaction interpreted as adipose tissue. This limitation is even more emphasized as the tile-based approach gives coarse WSI segmentations. Coarse representations can be used for simple tasks such as locating tumor areas or checking for depth of invasion. Still, more is needed when computing detailed metrics such as tumor-to-stroma interaction. For such tasks, advanced segmentation models are required. Unfortunately, segmentation architectures often rely on pixel-wise annotations to be trained, which are scarce and tedious to acquire.

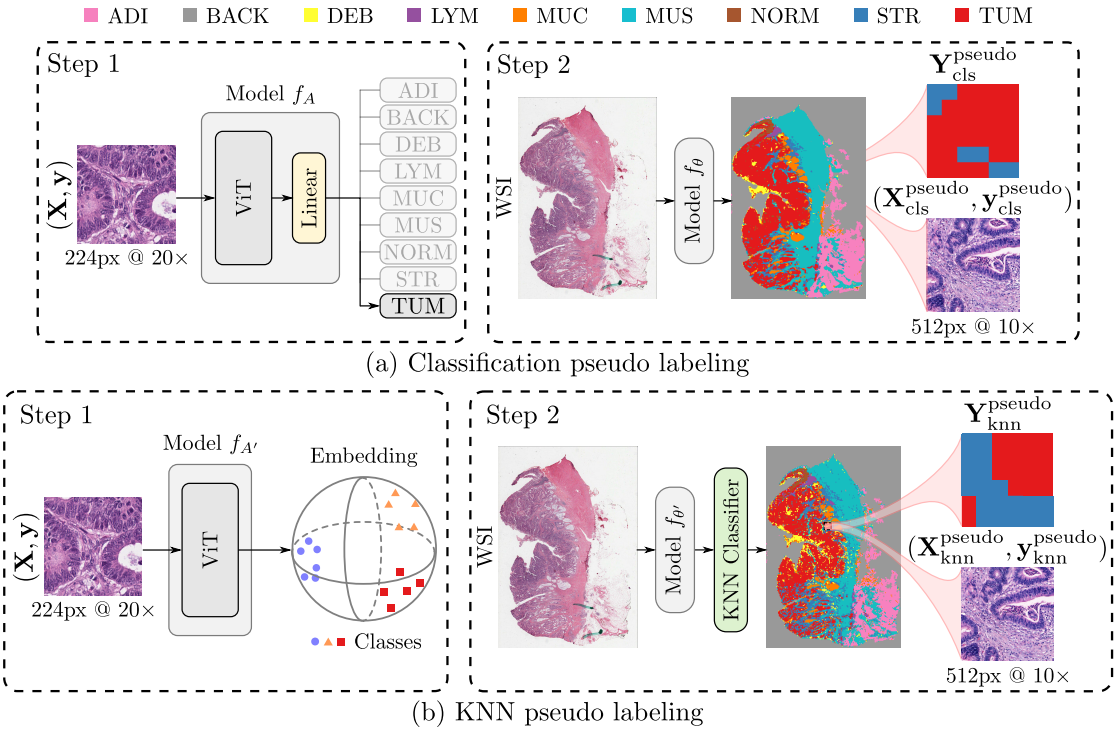
In the next chapter, we use self-supervised learning to improve tissue segmentation.

5 Coarse to Refined: Improving Tissue Detection

In the previous chapter, we built an adaptation framework that uses weakly labeled source sets to classify target whole slide image (WSI) patches. Such classification architectures rely on sliding window approaches that produce coarsely segmented output at inference time. These representations are not suited for medical applications where fine-grained segmentation is needed. In addition, publicly available annotated data are often designed for classification tasks [77, 79, 71] and are composed of homogeneous tissue examples lacking contextual information. For example, distinguishing between stroma and smooth muscle is a non-trivial task when surrounding tissues are unknown. Our motivation is to combine self-supervised learning (SSL) and weakly supervised semantic segmentation (WSSS) to segment complex tissue structures efficiently. WSSS is defined as the use of weakly labeled data (*e.g.* patch label) to train segmentation models. This setting is more convenient than supervised segmentation tasks requiring pixel-wise annotation.

In this chapter, we propose our coarse to refined (C2R) data-efficient training approach. The method tackles previous observations on data heterogeneity and output resolution. Furthermore, we assume that collecting dense pixel-labeled data is an uphill task. While segmentation annotation at low magnification is straightforward, generating precise dense pixel annotation at high resolution (*i.e.*, gigapixel histology images) is tricky. The annotation quality highly relies on the staining or potential local artifact and is prone to error. As a result, we avoid using extra manually annotated data as much as possible.

In the first step, we train a shallow network to classify tissue patches using open-source labeled data. The model is then used to generate coarse pseudo labels from WSIs (section 5.1), thus creating a large bank of tissue representations. In the second step, we use the generated pseudo labels to train a segmentation network (section 5.2) to refine the detection of the tissue while taking advantage of the contextual information and visual consistency. Then, we propose a novel way to validate our results while avoiding additional human annotation by using staining information (section 5.3). Finally, we perform ablation studies of the presented architecture on diverse datasets (section 5.4).



(a) Classification pseudo labeling

(b) KNN pseudo labeling

Figure 5.1 – Creation of pseudo labels. (a) Classification approach where we use a model f_A and apply it on WSI tiles to create pseudo labels $\mathbf{Y}_{\text{cls}}^{\text{pseudo}}$, $\mathbf{y}_{\text{cls}}^{\text{pseudo}}$, and $\mathbf{X}^{\text{pseudo}}$. (b) k-nearest neighbors (KNN) approach where we create a feature embedding from $f_{A'}$ to assign pseudo labels $\mathbf{Y}_{\text{knn}}^{\text{pseudo}}$, $\mathbf{y}_{\text{knn}}^{\text{pseudo}}$, and $\mathbf{X}^{\text{pseudo}}$.

5.1 Pseudo Labeling

In this section, we tackle the creation of pseudo labels. As previously explained, acquiring pixel-wise labels for segmentation is a tedious task. Instead, we propose a simple approach to generate pseudo labels for our C2R model. Firstly, we build a shallow architecture to perform coarse tissue classification using publicly available data. Secondly, we apply the model to WSIs to create large amounts of pseudo-labeled data. By sampling directly from WSIs, we can ensure data heterogeneity (*i.e.* mixture of tissue classes). Consequently, we aim to take advantage of WSI complex structures to improve tissue representation without additional annotations. In Figure 5.1, we depict an overview of the pseudo label creation.

More formally, we want to be able to generate various annotated tissue regions from WSIs as $(\mathbf{X}^{\text{pseudo}}, \mathbf{y}^{\text{pseudo}})$. Here $\mathbf{X}^{\text{pseudo}} \in \mathbb{R}^{W_p \times H_p \times 3}$ is a RGB tile with width W_p and height H_p. The variable $\mathbf{y}^{\text{pseudo}} \in [0, 1]^C$ is the pseudo label linked to the selected tile and represents the local distribution of classes in the image. We propose two solutions to compute the pseudo labels based on classification (subsection 5.1.1) and k-nearest neighbors (KNN) (subsection 5.1.2).

5.1.1 Classification

Our first approach relies on tissue classification. First, we train a model f_A to distinguish between C classes. The architecture comprises a pre-trained vision transformer (ViT) network with a linear classification head attached on top. The weights of the ViT backbone are fixed, and solely the head is updated through training. We feed the model RGB tiles $\mathbf{X} \in \mathbb{R}^{H \times W \times 3}$ as input and aim at predicting target $\mathbf{y} \in [0,1]^C$, where W, H represent the height and width of the tile. We suppose the tiles come from a publicly available dataset and thus are labeled.

Once trained, the model is applied on WSIs using a sliding window approach to get a first estimation of slide segmentations. We then randomly select a sub-region of the WSIs as $\mathbf{X}^{\text{pseudo}}$ as well as its corresponding classification map $\mathbf{Y}_{\text{cls}}^{\text{pseudo}} \in [0,1]^{w_p \times h_p \times C}$ over C classes where w_p, h_p represent the height and width of the classification map. We define the pseudo label of the given region $\mathbf{X}^{\text{pseudo}}$ as the average of class probabilities over the classification map:

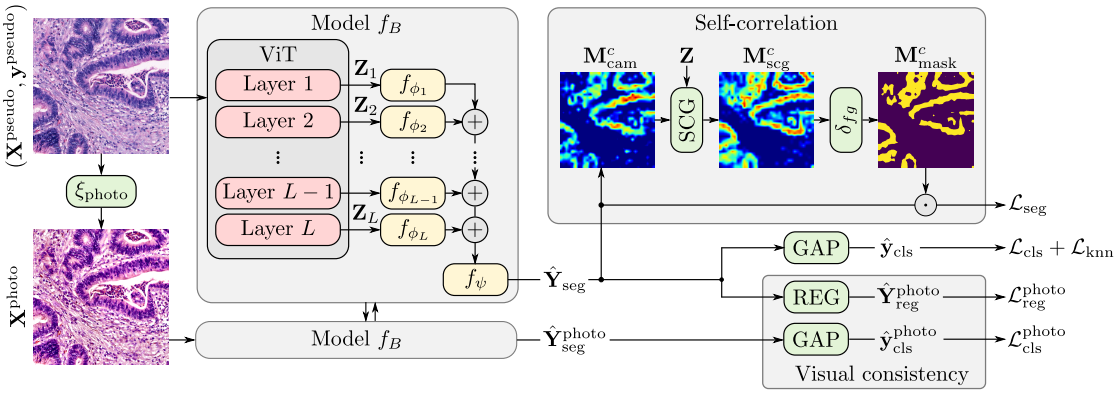
$$\mathbf{y}_{\text{cls}}^{\text{pseudo}} = \frac{1}{h_p w_p} \sum_{i=1}^{h_p} \sum_{j=1}^{w_p} \left(\mathbf{Y}_{\text{cls}}^{\text{pseudo}} \right)_{i,j} . \tag{5.1}$$

Here, i, j are the index of the matrix along the first two dimensions (namely height and width). The final pseudo label pair for the classification approach is given as $(\mathbf{X}^{\text{pseudo}}, \mathbf{y}_{\text{cls}}^{\text{pseudo}})$. Note that the width and height of the tile and its segmentation are proportional as $W_p \propto w_p$ and $H_p \propto h_p$.

5.1.2 k-Nearest Neighbors

For the second approach, we use a pre-trained ViT architecture $f_{A'}$. Here, we feed the model RGB tiles from publicly available datasets to get feature representations. Based on the generated embedding, we build a KNN classifier over the C classes. To get the prediction of a given WSI tile, we feed it to the pre-trained model, look for the k-nearest samples, and use majority voting for the final decision. Then, we apply the same logic as in the classification setup, where we randomly select a sub-region of the WSI as $\mathbf{X}^{\text{pseudo}}$ as well as its corresponding segmentation $\mathbf{Y}_{\text{knn}}^{\text{pseudo}} \in \{0,1\}^{h_p \times w_p \times C}$. We select the same subareas as the previous approach to have pseudo-label correspondence for the training in the next section. We define the pseudo label using the same approach:

$$\mathbf{y}_{\text{knn}}^{\text{pseudo}} = \frac{1}{h_p w_p} \sum_{i=1}^{h_p} \sum_{j=1}^{w_p} \left(\mathbf{Y}_{\text{knn}}^{\text{pseudo}} \right)_{i,j} . \tag{5.2}$$



Abbreviations: GAP (global average pooling), SCG (self-correlation map generating), REG (regulation)

Figure 5.2 – Proposed architecture for our coarse to refined (C2R) approach. We feed $\mathbf{X}^{\text{pseudo}}$, $\mathbf{y}^{\text{pseudo}}$ to our ViT backbone to generate $\mathbf{Z}_1, \ldots, \mathbf{Z}_L$ embeddings. The predictions are aggregated at every level using classification heads $f_{\phi_1}, \ldots, f_{\phi_L}, f_\psi$ to create the segmentation map $\hat{\mathbf{Y}}_{\text{seg}}$. The segmentation map is constrained by a self-correlation loss $\mathcal{L}_{\text{seg}}$, classification losses $\mathcal{L}_{\text{cls}}$, $\mathcal{L}_{\text{knn}}$, and visual consistency losses $\mathcal{L}^{\text{photo}}_{\text{reg}}$, $\mathcal{L}^{\text{photo}}_{\text{cls}}$. For the visual consistency, we generate an augmentation $\mathbf{X}^{\text{photo}}$ of the original image using transformations ξ_{photo} and fed it to the model to produce $\hat{\mathbf{Y}}^{\text{photo}}_{\text{seg}}$.

5.2 Method

In this section, we introduce our network as depicted in Figure 5.2. The main architecture $f_B : \mathbb{R}^{W_p \times H_p \times 3} \to \mathbb{R}^{W_s \times H_s \times C}$ takes as input a tile $\mathbf{X}^{\text{pseudo}} \in \mathbb{R}^{W_p \times H_p \times 3}$ and predict a segmentation map $\hat{\mathbf{Y}}_{\text{seg}} \in \mathbb{R}^{W_s \times H_s \times C}$. For training, we rely on the pseudo labels $\mathbf{y}^{\text{pseudo}} \in \mathbb{R}^C$ previously generated. For each input image, we have access to two pseudo label estimations as $\mathbf{y}^{\text{pseudo}}_{\text{cls}}$ and $\mathbf{y}^{\text{pseudo}}_{\text{knn}}$.

The image first goes through a ViT that outputs feature embeddings $\mathbf{Z}_l \in \mathbb{R}^{W_s \times H_s \times D}$, $l \in \{1, \ldots, L\}$ that are the representation of the input image at the l-th ViT layer and where D is the dimension of the ViT feature space. The embeddings are passed through a set of nonlinear classifier $f_{\phi_l} : \mathbb{R}^{W_s \times H_s \times D} \to \mathbb{R}^{W_s \times H_s \times C}$ with parameters ϕ_l to get a second representation where C is the number of classes. Note that the classifiers' weights are not shared across the architecture, as each layer embedding is unique. Finally, the predictions are concatenated and passed through a last classifier $f_\psi : \mathbb{R}^{W_s \times H_s \times L \cdot C} \to \mathbb{R}^{W_s \times H_s \times C}$ with parameters ψ to get the final segmentation map $\hat{\mathbf{Y}}_{\text{seg}}$. We define the overall loss as the contribution of multiple terms:

$$\min \mathcal{L}_{\text{C2R}} = \min_{\phi_1, \ldots, \phi_L, \psi} \left(\underbrace{\mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{knn}}}_{\text{classification}} + \underbrace{\mathcal{L}_{\text{seg}}}_{\text{segmentation}} + \underbrace{\mathcal{L}^{\text{photo}}_{\text{cls}} + \mathcal{L}^{\text{photo}}_{\text{reg}}}_{\text{visual consistency}} \right). \tag{5.3}$$

Algorithm 3: Pseudocode for the C2R framework.

Initialize ViT with pretrained weights and freeze them ;
Initialize thresholds δ_{fg} to constant ;
for e = 0 **to** $N_{\text{epochs}} - 1$ **do**
  for batch $\{(\mathbf{X}^{\text{pseudo}}, \mathbf{y}_{\text{cls}}^{\text{pseudo}}, \mathbf{y}_{\text{knn}}^{\text{pseudo}})_i\}_{i=1}^{B}$ **do**
    Get augmented samples $\mathbf{X}^{\text{photo}}$ using data augmentation ξ_{photo} ;
    Perform forward pass using ViT ;
    Reconstruct embedding $\mathbf{Z}$ using layers' outputs ; ▷ Equation 5.9
    Perform forward passes $f_{\phi_1}, \ldots, f_{\phi_L}, f_\psi$ to get segmentation maps $\hat{\mathbf{Y}}_{\text{seg}}$ and $\hat{\mathbf{Y}}_{\text{seg}}^{\text{photo}}$;
    Compute self-correlation refinement $\mathbf{M}_{\text{scg}}^c$; ▷ Equation 5.12
    Get refined foreground mask $\mathbf{M}_{\text{mask}}^c$; ▷ Equation 5.13
    Compute classification loss $\mathcal{L}_{\text{cls}}$ and $\mathcal{L}_{\text{knn}}$; ▷ Equation 5.5 - 5.6
    Compute segmentation loss $\mathcal{L}_{\text{seg}}$; ▷ Equation 5.14
    Compute visual consiftency $\mathcal{L}_{\text{cls}}^{\text{photo}}$ and $\mathcal{L}_{\text{reg}}^{\text{photo}}$; ▷ Equation 5.16 and 5.22
    Evaluate overall loss $\mathcal{L}_{\text{C2R}}$; ▷ Equation 5.3
    Update thresholds δ_{fg} ; ▷ Equation 5.15
    Update weights $\phi_1, \ldots \phi_L, \psi$ with backpropagation ;
  end
end

First, we use classification losses from the previously generated pseudo labels. Secondly, we use a segmentation loss based on self-correlation to refine the prediction of classes. Finally, we add visual consistency terms to increase the robustness of the predictions under stain variation and artifacts. The ViT weights are fixed, and the optimization is performed over the sets of parameters $\phi_1, \ldots, \phi_L$, and ψ. We describe the implementation of classification losses in subsection 5.2.1, the segmentation loss in subsection 5.2.2, and visual consistency losses in subsection 5.2.3. The C2R pseudocode for the presented architecture is given in algorithm 3.

5.2.1 Classification

We define a classification loss between the output prediction and the previously generated pseudo labels. We use multilabel soft margin (MLSM) to compute errors on multi-labels as the loss is able to handle mixtures of class probability as target labels [73]. The MLSM function gives the error between a target $\mathbf{y}$ and prediction $\hat{\mathbf{y}}$:

$$\mathcal{L}_{\text{MLSM}}(\mathbf{y}, \hat{\mathbf{y}}) = -\frac{1}{C} \sum_{i=1}^{C} y_i \log\left(\frac{1}{1+\exp(-\hat{y}_i)}\right) + (1-y_i)\log\left(\frac{\exp(-\hat{y}_i)}{1+\exp(-\hat{y}_i)}\right). \quad (5.4)$$

We then define the losses as the error between the global average pooling (GAP) of the

predictions and pseudo labels $\mathbf{y}_{\text{cls}}^{\text{pseudo}}$ and $\mathbf{y}_{\text{knn}}^{\text{pseudo}}$ as:

$$\begin{aligned}\mathcal{L}_{\text{cls}} &= \mathcal{L}_{\text{MLSM}}\left(\mathbf{y}_{\text{cls}}^{\text{pseudo}}, \hat{\mathbf{y}}_{\text{cls}}\right) \\ &= \mathcal{L}_{\text{MLSM}}\left(\mathbf{y}_{\text{cls}}^{\text{pseudo}}, \frac{1}{W_s H_s}\sum_{i=1}^{W_s}\sum_{j=1}^{H_s}(\hat{\mathbf{Y}}_{\text{seg}})_{i,j}\right),\end{aligned} \tag{5.5}$$

$$\begin{aligned}\mathcal{L}_{\text{knn}} &= \mathcal{L}_{\text{MLSM}}\left(\mathbf{y}_{\text{knn}}^{\text{pseudo}}, \hat{\mathbf{y}}_{\text{cls}}\right) \\ &= \mathcal{L}_{\text{MLSM}}\left(\mathbf{y}_{\text{knn}}^{\text{pseudo}}, \frac{1}{W_s H_s}\sum_{i=1}^{W_s}\sum_{j=1}^{H_s}(\hat{\mathbf{Y}}_{\text{seg}})_{i,j}\right).\end{aligned} \tag{5.6}$$

5.2.2 Segmentation and Self-correlation

With the current setup, we do not impose any constraint on the segmentation map (*i.e.* output features). Here, we aim to generate a mask that estimates the class location within the feature map and use it to improve our segmentation. To do so, we employ the concept of class activation maps (CAMs) [162]. CAMs are local attentions of the model for a given class and image. A high value in a CAM is linked to a significant contribution of the area to the class prediction. As a result, CAMs are directly correlated with the presence of the class. It is defined as:

$$\mathbf{M}_{\text{cam}} = (\theta^{[L]})^{\top}\hat{\mathbf{Y}}_{\text{seg}}^{[L-1]}, \tag{5.7}$$

where $\theta^{[L]}$ are the weight of the last layer of the model and $\hat{\mathbf{Y}}_{\text{seg}}^{[L-1]}$ the model features evaluated at second to last layer. In our architecture, we use a nonlinear classification head f_{ψ} with no bias to create our segmentation output. Consequently, in our setup, the CAMs are implicitly defined:

$$\begin{aligned}\mathbf{M}_{\text{cam}} =& \hat{\mathbf{Y}}_{\text{seg}} \\ =& \begin{pmatrix}\hat{\mathbf{Y}}_{\text{seg}}^{1} & \dots & \hat{\mathbf{Y}}_{\text{seg}}^{C}\end{pmatrix},\end{aligned} \tag{5.8}$$

where, $\hat{\mathbf{Y}}_{\text{seg}}^{c} \in \mathbb{R}^{W_s \times H_s}$ is the CAM for class c. The CAMs give a coarse estimation of the model's activations. To refine our prediction, we use self-correlation map generating (SCG) [112]. The SCG compute the first-degree correlation between the extracted features of the model and sum their contributions for a given class mask. Here, we consider the

output of the pre-trained model as the feature reference for the self-correlation. Each layer l of the ViT outputs a representation $\mathbf{Z}_l \in \mathbb{R}^{H_s W_s \times D}$ where $H_s W_s$ is the flattened spatial size of the feature map and D the dimension of the embedding. We create the overall image descriptor $\mathbf{Z}$, which is the concatenation of each layer embedding:

$$\begin{aligned}\mathbf{Z} &= \begin{pmatrix}\mathbf{Z}_1 & \cdots & \mathbf{Z}_L\end{pmatrix} \in \mathbb{R}^{H_s W_s \times LD} \\ &= \begin{pmatrix}\mathbf{f}_1 & \cdots & \mathbf{f}_{HsWs}\end{pmatrix}^\top .\end{aligned} \tag{5.9}$$

By looking at the transpose matrix, we can also see $\mathbf{Z}$ as a list of $H_s W_s$ feature descriptors $\mathbf{f}_i \in \mathbb{R}^{LD}$. The vector $\mathbf{f}_i$ is the embedding of the i-th entry of the segmentation map. We can now compute the first order self-correlation $\mathbf{S} \in \mathbb{R}_+^{HW \times HW}$ between the representation of the segmentation map entries:

$$\begin{aligned}S_{i,j} &= \max(0, \frac{\mathbf{f}_i^\top \mathbf{f}_j}{\|\mathbf{f}_i\| \|\mathbf{f}_j\|}), \\ \mathbf{S} &= (S_{i,j})_{1 \leq i,j \leq HW}.\end{aligned} \tag{5.10}$$

We use cosine similarity to measure the affinity between two feature entries. We can interpret $S_{i,j}$ as the correlation between the i-th segmentation map entry and its j-th element. A high value means that those two locations share similar content. We then use the CAMs to remove the background information from the correlation map and select the class-relevant objects:

$$\begin{aligned}\mathbf{U}^c &= \{\mathbf{S}\}_{(\mathrm{vec}(\hat{\mathbf{Y}}^c_{\mathrm{seg}})>0)} \\ &= \begin{pmatrix}\mathbf{u}_1^c & \cdots & \mathbf{u}_N^c\end{pmatrix},\end{aligned} \tag{5.11}$$

$$\mathbf{M}^c_{\mathrm{scg}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{u}_i^c \,. \tag{5.12}$$

We use the masking function $\{\mathbf{S}\}_A$ and condition A to select the columns of the self-correlation map that are part of the class activation and create a compressed representation of the segmentation $\mathbf{U}^c$. In other words, given the condition $A = \mathrm{vec}(\hat{\mathbf{Y}}^c_{\mathrm{seg}}) > 0$ and $A \in \{0,1\}^{H_s W_s}$ we select the columns of $\mathbf{S}$ where A is nonzero (*i.e.* class foreground map). The result is then averaged over the selected columns to create the final SCG map. As we rely on positive cosine similarity to compute self-correlation, the element of

the map $\mathbf{M}^c_{\text{scg}}$ are all contained in the range $[0, 1]$. They can be seen as the probability of each entry belonging to a given class c. We get the segmentation mask by applying a threshold $\delta_{\text{fg}} \in [0, 1]$ to the confidence score SCG:

$$\mathbf{M}^c_{\text{mask}} = \mathbf{M}^c_{\text{scg}} > \delta_{\text{fg}}. \tag{5.13}$$

The mask represents the refined foreground for a class c where we impose class consistency:

$$\mathcal{L}_{\text{seg}} = \mathcal{L}_{\text{MLSM}} \left(\mathbf{y}^{\text{pseudo}}, \{\hat{\mathbf{Y}}_{\text{seg}}\}_{(\mathbf{M}^c_{\text{mask}} \neq 0)} \right), \qquad \underset{c \in \{1,\dots,C\}}{\arg\max}\, \mathbf{y}^{\text{pseudo}}_c. \tag{5.14}$$

In practice, we compute the loss on the majority class. In this case, the masking function is performed over the first two dimensions of the matrix $\hat{\mathbf{Y}}_{\text{seg}}$. We apply the MLSM loss for every entry of the segmentation map that reaches the confidence threshold δ_{fg}. The threshold is a hyper-parameter of the model that can be tricky to estimate. In the literature, the threshold selection is often obtained empirically and fixed [73]. However, the optimal value is likely to change during training as the segmentation maps are estimated. Moreover, the threshold might vary depending on the selected class. Here, we propose to use a moving average (MVA) to update class-specific thresholds during the learning procedure. We define the update of the threshold δ^c_{fg} of a given class c:

$$\delta^c_{\text{fg}} \leftarrow m\, \delta^c_{\text{fg}} + \frac{(1-m)}{\sum_{i=1}^{H_s} \sum_{j=1}^{W_s} (\mathbf{M}^c_{\text{mask}})_{i,j}} \left(\mathbf{M}^c_{\text{scg}} \odot \mathbf{M}^c_{\text{mask}} \right). \tag{5.15}$$

where $\odot$ is the element-wise product and $m \in [0, 1]$ the update momentum.

5.2.3 Visual Consistency

Aside from classification and segmentation losses, we also impose visual consistency. The idea is that for a given input image $\mathbf{X}^{\text{pseudo}}$ we can generate an augmented image $\mathbf{X}^{\text{photo}}$ that shares a similar structure with the input using a set of transformation $\xi_{\text{photo}} : \mathbb{R}^{H_p \times W_p \times 3} \rightarrow \mathbb{R}^{H_p \times W_p \times 3}$. We select the transformations such that they only affect the visual aspect of the image but not its geometry (*e.g.* colorspace shift). As a result, overlapping the two images still produces pixel-wise matching between the two entries. We define the estimated segmentation map of the augmented view as $\hat{\mathbf{Y}}^{\text{photo}}_{\text{seg}} = f_B(\mathbf{X}^{\text{photo}})$. We follow the same logic as before, where we average the prediction across the segmentation map to get the final class probability. The visual consistency

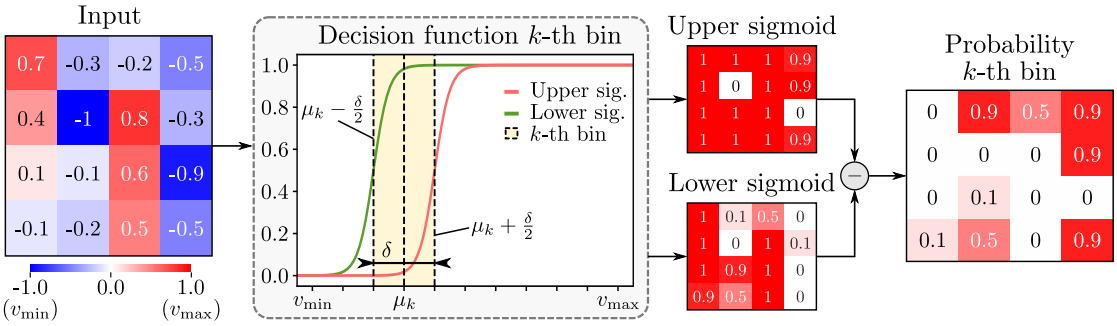


Figure 5.3 – Probability of given feature map samples in range $[v_{min}, v_{max}]$ to belong to the k-th bin with center c_k and width δ. We use two sigmoid functions as upper and lower bounds. The final statistics are given as the difference between the two decision functions. In this example we assume $v_{max} = -v_{min} = 1$, $K = 10$, and $k = 3$.

classification loss is given as follows:

$$\begin{aligned}\mathcal{L}_{\text{cls}}^{\text{photo}} &= \mathcal{L}_{\text{MLSM}}\left(\mathbf{y}_{\text{cls}}^{\text{pseudo}}, \hat{\mathbf{y}}_{\text{cls}}^{\text{photo}}\right) \\ &= \mathcal{L}_{\text{MLSM}}\left(\mathbf{y}_{\text{cls}}^{\text{pseudo}}, \frac{1}{W_s H_s}\sum_{i=1}^{W_s}\sum_{j=1}^{H_s}(\hat{\mathbf{Y}}_{\text{seg}}^{\text{photo}})_{i,j}\right).\end{aligned} \tag{5.16}$$

Another way to take advantage of the augmented frame is to link the feature space of the original view and its augmented version in a self-supervised fashion. We use l_1-norm as in [72] to minimize the distance between the feature spaces:

$$\mathcal{L}_{\text{reg}}^{\text{photo}} = \|\max(\hat{\mathbf{Y}}_{\text{seg}}, 0) - \max(\hat{\mathbf{Y}}_{\text{seg}}^{\text{photo}}, 0)\|_1 . \tag{5.17}$$

However, l_1-norm assumes we have a strict one-to-one correspondence between the features, which can be too constraining. Another option is to match the feature distribution between the two domains using histograms instead of relying on element-wise prediction. Unfortunately, the histogram function itself is not differentiable. To overcome the issue, we use shifted sigmoids to estimate feature distributions [144]. The process is fully differentiable and is depicted in Figure 5.3. We define as μ_k the centers of the histogram bins with $k \in \{1, \ldots, K\}$:

$$\mu_k = v_{\text{min}} + (k + 0.5)\delta \qquad \text{and} \qquad \delta = \frac{v_{\text{max}} - v_{\text{min}}}{K}. \tag{5.18}$$

The values v_{max} and $v_{\text{min}} \in \mathbb{R}$ are the range of the histogram, K the number of bins, and

δ their widths. For each histogram bin, we use a sigmoid function s_λ to compute the probability vector's elements $\mathbf{z} = \begin{pmatrix} z_1 & \dots & z_D \end{pmatrix}$ to belong to the cluster μ_k:

$$s_\lambda(z) = \frac{1}{1 + \exp(-\lambda z)}, \tag{5.19}$$

$$r_k(\mathbf{z}) = \frac{\sum_{d=1}^{D} s_\lambda((z_d - \mu_k) + \frac{\delta}{2}) - s_\lambda((z_d - \mu_k) - \frac{\delta}{2})}{\sum_{l=1}^{K}\sum_{d=1}^{D} s_\lambda((z_d - \mu_l) + \frac{\delta}{2}) - s_\lambda((z_d - \mu_l) - \frac{\delta}{2})}, \tag{5.20}$$

where $\lambda \in \mathbb{R}$ is a hyperparameter that fixes the sharpness of the sigmoid and D the dimension of the vector. The larger the value λ, the more selective the sigmoid function. We define the reference $\mathbf{P} = (q_{c,k})_{1 \le c \le C, 1 \le k \le K}$ and target $\mathbf{Q} = (q_{c,k})_{1 \le c \le C, 1 \le k \le K}$ distributions:

$$p_{c,k} = r_k(\mathrm{vec}(\mathbf{Y}^c_{\mathrm{seg}})) \qquad \text{and} \qquad q_{c,k} = r_k(\mathrm{vec}(\mathbf{Y}^{\mathrm{photo},c}_{\mathrm{seg}})). \tag{5.21}$$

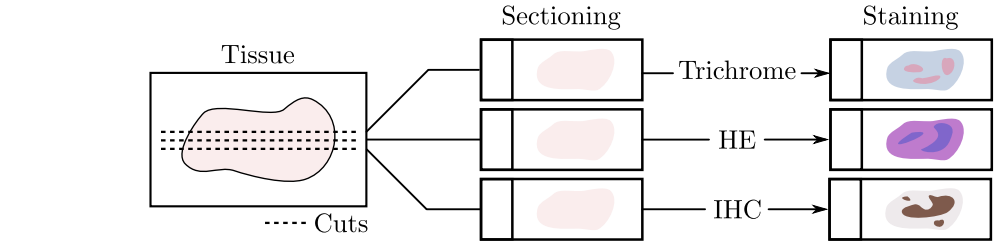
The histogram matching is performed across each individual class. The loss is given as the Kullback-Leibler (KL) between the estimated feature distributions of the reference segmentation map $\mathbf{Y}_{\mathrm{seg}}$ and its augmented view $\mathbf{Y}^{\mathrm{photo}}_{\mathrm{seg}}$.

$$\mathcal{L}^{\mathrm{photo}}_{\mathrm{reg}} = \mathrm{KL}(\mathbf{P} \| \mathbf{Q}) = \sum_{c=1}^{C}\sum_{k=1}^{K} p_{c,k} \log \frac{p_{c,k}}{q_{c,k}}. \tag{5.22}$$

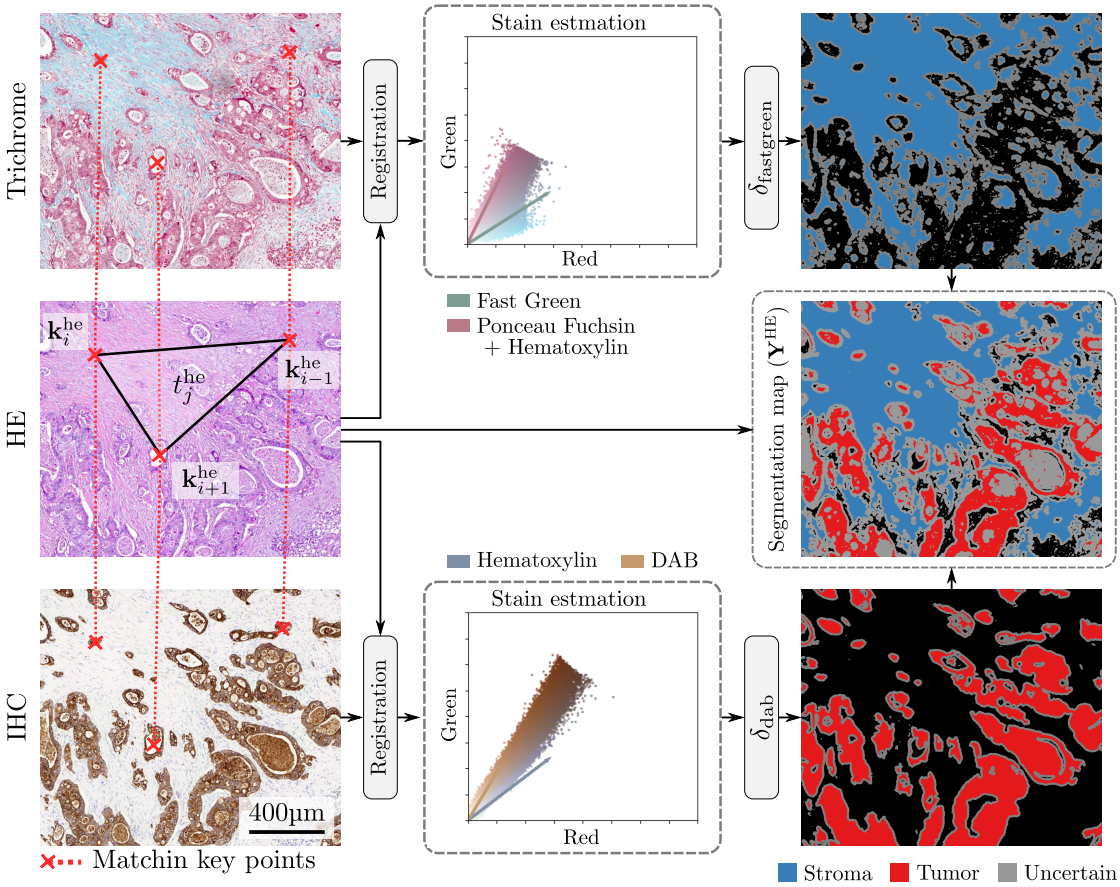
5.3 Use Staining as Validation

In the previous section, we introduced our WSSS approach to improve segmentation from coarse labels. To evaluate the performance of our method, we need pixel-wise ground truth annotations from colorectal cancer (CRC) tissue, which are unavailable. Manual segmentation is a tedious task requiring the annotator to follow tissue boundaries across gigapixel images carefully. Moreover, certain areas are not clearly separable as they are, in fact, mixtures of classes. This phenomenon is worsened by the use of HE staining that can create poor contrast between certain classes. Hence, even for a trained pathologist, distinguishing between tissue types is challenging and time-consuming at high magnification.

We propose a novel and fast way to generate segmentation labels from WSIs without the need for tiresome annotation. More precisely, we focus on tumor and stroma detection as



(a) Tissue consecutive cuts and stainings



(b) Tumor and stroma estimation from stained images

Figure 5.4 – Generating segmentation labels from stained WSIs. (a) Process of tissue cutting and staining into Trichrome, hematoxylin and eosin (HE), and immunohistochemistry (IHC). (b) Registration of Trichrome and IHC to HE slide using matching key points $\mathbf{k}_i^{\text{he}}$ and triangles t_j^{he}. Stains are extracted from the registered slides, thresholded using $\delta_{\text{fastgreen}}$ and δ_{dab}, and merged to create the segmentation map.

their interaction has been proven to be an interesting prognostic factor [141, 138]. Given a fixed tissue specimen from a patient, we cut three consecutive slides as depicted in Figure 5.4a. The top slide is stained using Trichrome, highlighting the smooth muscle in red/purple and collagen fibers (*i.e.* stroma) in green/blue. The center cut is stained

using the standard HE procedure to color basic and acidic components. Finally, the bottom slide is stained with IHC to enhance the presence of tumor cells in brown.

As the acquired slides are from consecutive cuts, we assume that local changes in shapes and textures are negligible. However, WSIs are vast. The addition of local small displacement due to tissue steering can result in large offsets between images at the WSI level. To tackle this issue, we consider the central HE slide as the reference and use it to register the Trichrome and IHC images. Image alignment can be performed using automatic feature extractors [83, 121]. However, automated approaches tend to fail when applied to histological images as WSIs have poor local feature descriptors. Another solution is to use homography. For each image, we manually select N paired points $\{(\mathbf{k}_i^{\text{trichrome}}, \mathbf{k}_i^{\text{he}}, \mathbf{k}_i^{\text{ihc}} \in \mathbb{R}^2)\}_{i=1}^{N}$ that matches across the three cuts. The number of selected locations is typically low to keep the annotation process simple and fast. We then use Delaunay triangulation [41] to build triangle meshes out of the generated key points.

Let's assume we end up with M matching triangles between images. For each triangle t_j^{ihc} in the IHC image, we have a corresponding triangle t_j^{he} in the reference HE image, where $j \in \{1, \dots, M\}$ denotes the index of the triangle. We build a projection matrix that matches the triangle source point to the reference one. The projection matrix is then applied to the source triangle to deform it locally. We apply the same approach to match the triangles $t_j^{\text{trichrome}}$ of the Trichrome image to the reference HE image. The procedure is depicted in Figure 5.4b.

Once registered, we estimate the staining distribution using the Macenko [96] approach. For the Trichrome image, we extract the Fast Green component as $\mathbf{M}^{\text{trichrome}} \in \mathbb{R}^{H \times W}$ that is correlated with the presence of collagen and therefore stroma. On the IHC slide, we isolate the diaminobenzidine (DAB) component as $\mathbf{M}^{\text{ihc}} \in \mathbb{R}^{H \times W}$ that highlight tumor cells. We carefully selected slides that do not include normal tissue as epithelial cells would appear in the DAB channel and be mixed with the tumor. We apply threshold $\delta_{\text{fastgreen}}$ and δ_{dab} on Trichrome and IHC maps to generate the foreground masks. The prediction maps for the HE image $\mathbf{Y}^{\mathbf{he}} = (y_{i,j})_{1 \leq i \leq H, 1 \leq j \leq W}$ is defined as:

$$y_{i,j} = \begin{cases} \text{Stroma} & \text{, if } (\mathbf{M}^{\text{trichrome}})_{i,j} \geq \delta_{\text{fastgreen}} \text{ and } (\mathbf{M}^{\text{ihc}})_{i,j} < \delta_{\text{dab}} \\ \text{Tumor} & \text{, if } (\mathbf{M}^{\text{trichrome}})_{i,j} < \delta_{\text{fastgreen}} \text{ and } (\mathbf{M}^{\text{ihc}})_{i,j} \geq \delta_{\text{dab}} \\ \text{Uncertain} & \text{, if } (\mathbf{M}^{\text{trichrome}})_{i,j} \geq \delta_{\text{fastgreen}} \text{ and } (\mathbf{M}^{\text{ihc}})_{i,j} \geq \delta_{\text{dab}} \\ \text{Background} & \text{, otherwise.} \end{cases} \tag{5.23}$$

We add a small uncertainty margin around the detection maps to create a smooth transition between the classes. The overall procedure to generate segmentation labels

Algorithm 4: Pseudocode for segmentation label acquisition.

```
Cut three consecutive slides from a tissue block ;
Stain top, middle and bottom slides using trichrome, HE, and IHC, respectively ;
Select a set of N paired points (k_i^trichrome, k_i^he, k_i^ihc) that match between the slides ;
Apply Delaunay triangulation to build M matching triangle sets across images ;
for triangles tuple (t_j^trichrome, t_j^he, t_j^ihc) j ∈ {1, ..., M} do
    Find transformation from t_j^trichrome to t_j^he ;
    Apply transformation to trichrome triangle to align it ;
    Find transformation from t_j^trichrome to t_j^he ;
    Apply transformation to IHC triangle to align it ;
end
Extract Fast Green stain from Trichrome image ;
Extract DAB stain from IHC image ;
Apply threshold δ_fastgreen and δ_dab to prediction maps for stroma and tumor respectively ;
Merge maps and set as uncertain overlapping predictions ;
Create uncertainty margin around classes to allow smooth transition ;
```

from the consecutive cuts is summarized in algorithm 4.

5.4 Experiments

In the experiments section, we first detail the experimental setup for extracting the pseudo labels, the training of the segmentation architecture, and the acquisition of segmentation labels for validation in subsection 5.4.1. Then, we perform ablation studies on our in-house data in subsection 5.4.2 and show comparisons with our previous Self-Rule to Multi Adapt (SRMA) work. In subsection 5.4.3, we discuss the architecture's performance when applied to data from different scanners. Finally, in subsection 5.4.4, we further validate our approach using data from the SemiCol segmentation challenge.

5.4.1 Experimental Settings

As a reference for the pseudo labeling extraction, we use data from Kather 19 (K19) which is composed of $C = 9$ classes as adipose (ADI), background (BACK), debris (DEB), lymphocytes (LYM), muscle (MUS), mucin (MUC), normal mucosa (NORM), stroma (STR) and tumor (TUM). The size of the training input images is $H = W = 224$px at $20\times$ magnification (*i.e.* 0.4856µm/px). For the pseudo labeling, we use a pre-trained ViT model from Dino [23] whose weights are kept fixed. We use ViT-S/16 architecture with a feature space size of $D = 384$. On top of the ViT, we attach a linear classifier to create f_A. For $f_{A'}$, we directly use the output of the pre-trained model to fit the KNN classifier with $k = 5$. The output resolution of the classification is 109µm ($224\text{px} \cdot 0.4856\text{µm/px}$).

To generate our pseudo labels, we apply the trained models on slides from the patient set $\mathcal{P}_A$. We select $H_p = W_p = 512$px for the generated pseudo labeled at $10\times$ magni-

fication (*i.e.* 0.9712µm/px). The resolution of the pseudo label classification is hence 508µm (512px $\cdot$ 0.9712µm/px) to include contextual information. Out of the generated classification map on patient set $\mathcal{P}_A$, we randomly sample areas such that we end up with $270,000$ pseudo labels balanced over all classes ($30,000$ examples per class). The dataset is named $\mathcal{D}_{\text{C2R-WSI}}$ and is used train the C2R architecture.

For the segmentation, we use the same pre-trained ViT model with fixed model weights. The selected architecture is composed of a succession of $L = 12$ layers. We attach nonlinear classifiers f_{ϕ_l} to ViT layers $l = \{3, 6, 9, 12\}$ to build the intermediate feature representation [160]. All four nonlinear classifiers are composed of two fully connected layers with rectified linear unit (ReLU) activation, input and hidden layer dimension set to $D = 384$ and output layer to the number of classes C. Finally, the features are aggregated and passed through a final nonlinear classification stage f_ψ with input size $4C$, hidden dimension $4C$, and output size C. The resolution of the output is defined by the size of the ViT model patches (*i.e.* 16 pixels). Given the input size of the pseudo labels $H_p = W_p = 512$, we end up with $H_s = W_s = 32$ and a resolution of the segmentation map of 15.5µm. It represents an upscaling factor of 7 compared to the initial resolution of f_A. The architecture is trained for a single epoch with Adam optimizer, learning rate $lr = 5 \cdot 10^{-3}$, and weights decay $w = 5 \cdot 10^{-4}$

To generate photo samples, we use a set of transformations ξ_{photo} composed of color jittering, random gamma, ISO noise, coarse dropout, Gaussian noise, gray conversion, Gaussian blur, image compression, and contrast limited adaptive histogram equalization (CLAHE). The previously mentioned transformations do not affect the inner structure of the tissue. In addition, we use another set of transformations that alter the samples' geometry. It comprises random resized crop, horizontal flip, rotation, and grid distortion. The transformations are applied simultaneously to both images to ensure feature correspondence. The mentioned transformations are available in the Python package Albumentations [19].

For the segmentation branch, we use $m = 0.99$ when updating thresholds to ensure slowly shifting values. If not specified otherwise, we set $\delta_{\text{fg}} = 0.05$ as starting value for all classes. When computing histograms for visual consistency, set the range of the histogram as $v_{max} = -v_{min} = \max(Q_{0.99}(\hat{\mathbf{Y}}_{\text{seg}}), Q_{0.99}(\hat{\mathbf{Y}}_{\text{seg}}^{\text{photo}}))$ to reduce the impact of the outlier. The number of bins and the scaling factor are empirically set to $K = 32$ and $\lambda = v_{max}/K$, respectively.

To generate segmentation labels, we use 7 tissue blocks from 7 different patients. The samples are selected to represent different cancer stages and depths of invasion. The tissues are all digitized using two scanners with different optics. In the result section, we refer to the scanners as A and B. If not mentioned, the results are given for images digitized using scanner B. Out of the selected tissue blocks, we extract 14 regions of interest (ROIs) to create segmentation maps. On average, we use $N = 11$ registration

Table 5.1 – Ablation study of loss term $\mathcal{L}_{\text{c2r}}$ and evaluation on $\mathcal{D}_{\text{C2R-ROI}}$ from scanner B. Results are averaged over 10 runs. We use an unpaired t-test with respect to the top result. The single level setting use only last ViT layer and multilevel use layers $\{3, 6, 9, 12\}$. For self-correlation we use different thresholds as $\delta_{\text{fg}} = \{0.05, 0.2, \text{MVA}\}$. We report class-wise F_1 score and Macro-F_1.

	C2R losses					TUM	STR	All
Methods	$\mathcal{L}_{\text{cls}}$	$\mathcal{L}_{\text{knn}}$	$\mathcal{L}_{\text{cls}}^{\text{photo}}$	$\mathcal{L}_{\text{reg}}^{\text{photo}}$	$\mathcal{L}_{\text{seg}}$	F_1	F_1	Macro-F_1
Classification								
f_θ						64.3 ± 2.5	75.6 ± 8.4	70.0 ± 4.0
SRMA [2]						68.1 ± 0.3	70.1 ± 2.3	69.1 ± 1.2
Segmentation (f_B)								
Single level								
C	✓					70.4 ± 1.4	71.7 ± 4.8	71.1 ± 2.7
K		✓				72.2 ± 0.7	66.1 ± 2.6	69.2 ± 1.4
CK	✓	✓				71.7 ± 1.7	73.9 ± 3.8	72.8 ± 2.3
Multilevel								
ML-C	✓					75.1 ± 3.1	80.6 ± 6.1	77.8 ± 4.1
ML-K		✓				75.0 ± 2.1	70.9 ± 4.8	72.9 ± 3.3
ML-CK	✓	✓				76.5 ± 2.7	79.4 ± 5.5	78.0 ± 3.2
Visual consistency								
ML-C-P	✓	✓	✓			75.2 ± 3.3	83.7 ± 2.8	79.4 ± 1.6
ML-C-R_{L1}	✓	✓		L1		**79.2** ± 1.5	**83.6** ± 3.9	**81.4** ± 2.4
ML-C-R_{H}	✓	✓		H		76.9 ± 2.6	82.6 ± 2.9	**79.7** ± 2.1
ML-C-PR_{L1}	✓	✓	✓	L1		**79.5** ± 1.8	**85.5** ± 2.8	**82.5** ± 2.0
ML-C-PR_{H}	✓	✓	✓	H		76.4 ± 2.8	83.3 ± 2.1	**79.8** ± 2.1
Self-correlation								
ML-C-$S_{0.05}$	✓	✓			0.05	**78.8** ± 3.3	**84.3** ± 4.7	**81.5** ± 3.8
ML-C-$S_{0.20}$	✓	✓			0.20	77.5 ± 2.6	83.0 ± 4.2	**80.3** ± 2.9
ML-C-S_{MVA}	✓	✓			MVA	**78.4** ± 2.2	**84.4** ± 3.3	**81.4** ± 2.6
Visual & Self-correlation								
ML-C-P-$S_{0.05}$	✓	✓	✓		0.05	**79.2** ± 2.1	**84.6** ± 2.7	**81.9** ± 2.0
ML-C-PR_{L1}-$S_{0.05}$	✓	✓	✓	L1	0.05	**80.0** ± 2.1	**84.5** ± 3.2	**82.2** ± 2.5
ML-C-PR_{H}-$S_{0.05}$	✓	✓	✓	H	0.05	**79.4** ± 2.0	**83.3** ± 3.6	**81.4** ± 2.5

Abbreviations: $\mathcal{L}_{\text{cls}}$ (C), $\mathcal{L}_{\text{knn}}$ (K), $\mathcal{L}_{\text{cls}}^{\text{photo}}$ (P), $\mathcal{L}_{\text{reg}}^{\text{photo}}$ (R), $\mathcal{L}_{\text{seg}}$ (S), Multilevel (ML), l^1-norm (L1), l^1-norm on foreground values (L1^+), histogram matching (H).

points across the ROIs. The threshold values for foreground selection in the optical density (OD) space are manually selected as $\delta_{\text{fastgreen}} \in [0.60, 0.90]$ and $\delta_{\text{dab}} \in [0.02, 0.06]$. We apply post-processing on the segmentation map to remove objects with an area smaller than the output segmentation resolution (*i.e.* 15µm × 15µm). The dataset is named $\mathcal{D}_{\text{C2R-ROI}}$ and used for validation.

5.4.2 Ablation Study - In-House Segmentation

We perform an ablation study of our proposed approach on our in-house $\mathcal{D}_{\text{C2R-ROI}}$ data. We use the segmentation labels generated using Trichrome and IHC staining as ground

truth. The models are applied on the central HE image. We report class-wise F_1 score and macro averaging F_1 on STR and TUM. Tissues detected by our architecture that are not part of either STR and TUM are considered incorrect. The results are presented in Table 5.1. We use the pseudo labeling model f_A as a baseline. For the losses, we use the abbreviations $\mathcal{L}_{\text{cls}}$ (C), $\mathcal{L}_{\text{knn}}$ (K), $\mathcal{L}_{\text{cls}}^{\text{photo}}$ (P), $\mathcal{L}_{\text{reg}}^{\text{photo}}$ (R) and $\mathcal{L}_{\text{seg}}$ (S). Regarding the ViT, we define the single level setting as the use of the last layer $l = 12$ and multilevel as the combined outputs of layers $l = \{3, 6, 9, 12\}$. For the visual consistency regulation term, we try l_1-norm (L1) and histogram matching (H) losses. About the self-correlation, we test different thresholds as $\delta_{\text{fg}} = \{0.05, 0.2, \text{MVA}\}$.

From the results, we can observe that using pseudo labels generated with the classification approach gives the best results. The use of KNN pseudo labels gives reasonable results for tumor detection but dramatically drops when used for stroma identification. This is due to the fact that the model interprets muscle tissue as stroma. This issue is partially solved when both losses are combined (CK), but not enough to achieve statistical improvement. On the other hand, multilevel architecture greatly improves the performance on both tumor and stroma with respect to the single-level stetting.

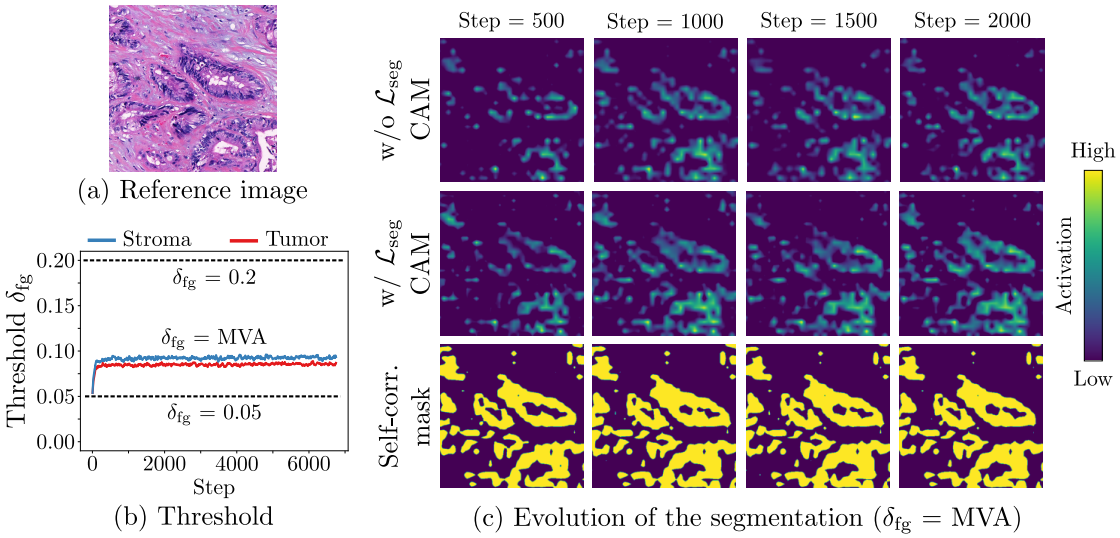
Regarding visual consistency, using an additional classification constraint (P) does not significantly improve the model's performance. Overall, combining both loss terms (PR) and l_1-norm produces better results. Self-correlation improves the performance of the model across all modalities. Regarding the threshold, keeping a lower threshold helps the model generate confident predictions. We observe a gain of +12% in tumor and stroma detection compared to our previous SRMA work.

Self-correlation Threshold

We go more in-depth about the selection of the foreground threshold. As detailed in the method, the threshold δ_{fg} indicates the confidence in the foreground map. Any value from the CAMs above the mentioned threshold is considered part of the class foreground mask.

In Figure 5.5, we show the behavior of the image thresholding for the self-correlation map. As a reference, we selected an image that includes both stroma and tumor. Moreover, we highlight the evolution of the threshold when using the MVA. Finally, we present how the tumor foreground mask map evolves with and without self-correlation loss for the MVA case.

We observe that the MVA stabilize after a few steps. This phenomenon is visible in the self-correlation where the target mask remains fixed. Another interesting observation is the behavior of the CAM with and without using the self-correlation loss. The use of $\mathcal{L}_{\text{seg}}$ forces the model to expand the detection and allow it to learn new feature representation. Consequently, the CAM merges features into more coherent and denser areas. However,



(a) Reference image

(b) Threshold

(c) Evolution of the segmentation (δ_{fg} = MVA)

Figure 5.5 – Evolution of the CAM with and without the presence of self-correlation constrain $\mathcal{L}_{seg}$. (a) Reference image with the presence of tumor tissue. (b) Variation of the thresholds $\delta_{fg} = \{0.05, 0.2, \text{MVA}\}$ through the training. (c) CAM without (top row) and with (middle row) self-correlation for different training steps and $\delta_{fg} = \text{MVA}$. The target mask for self-correlation is given in the last row.

when focusing on the numerical result, we do not see a difference between the use of a fixed threshold at $\delta_{fg} = 0.05$ or the use of the MVA. We prefer using MVA as no manual thresholding is required.

Histogram Matching

In our work, we propose two different approaches for regularizing visual consistency. The first approach relies on l^1-norm, while the second uses histogram matching. Here, we focus on the second regularization approach.

In Figure 5.6, we present an example of the evolution of the matching of histograms through the training. Each column shows a different training step, while each row represents a different training setting. With the top row, we display the case where no regulation is used. On the bottom row, we apply regularization as histogram matching. We highlight the real feature distribution (continuous line) as well as the estimated feature distribution using the proposed approach (blue/orange bins). We define as target the feature generated from the original view and as source the features of the augmented view.

We observe that the estimated and real feature distributions properly align across all steps in the regularized setting. As the training progresses, the regularization term can impose and maintain feature alignment between the source and target histogram. For the

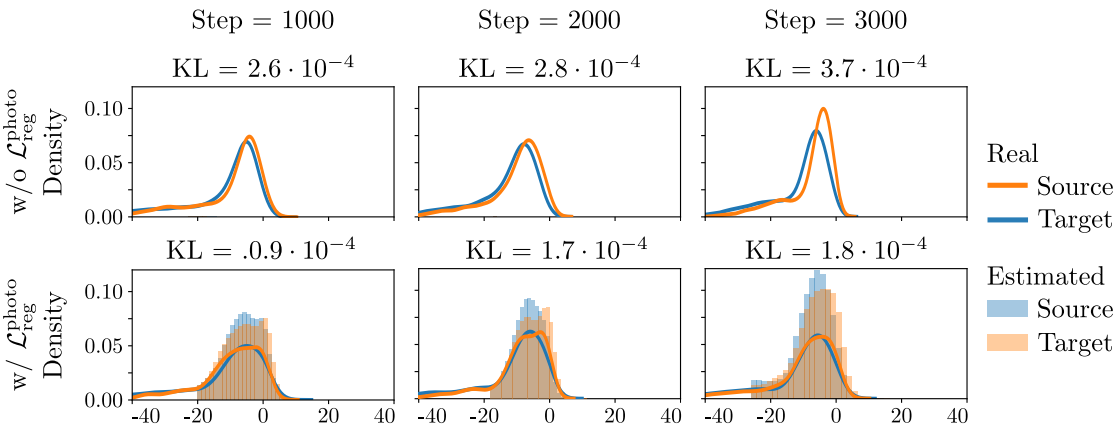


Figure 5.6 – The behavior of the histogram matching through training. Each column represents a different training step. The top and bottom rows show the baseline and the use of histogram matching for regularization loss $\mathcal{L}_{reg}^{photo}$, respectively. For the baseline, we show the real feature distribution for a given class. We denote as target the original image and as source the augmented view. For the histogram matching, we add the estimated feature distribution. For all plots, we report KL loss.

baseline, we observe that the two distributions shift after a few steps. The phenomenon is more visible with the computation of the KL divergence loss that increases as training progresses. The presented results show that the features are not centered around the boundary decision 0. Our previous assumption $v_{max} = -v_{min}$ is not optimal as the positive bins are less populated. However, as the histogram range is based on the upper bound, we ensure the range of positive values is properly represented.

Use Case - WSI Segmentation

In our previous SRMA work, a sliding window is used to classify local patches. This approach produces a coarse segmentation output that is unsuitable for clinical applications. In Figure 5.7, we visually compare the performance of our proposed C2R to our previous SRMA architecture. The C2R framework is able to generate fine-grained segmentation maps. The arrows highlight locations where small tissues are accurately identified. We can observe tumor-associated stroma that lies between tumor aggregates. These features are critical when evaluating various clinical metrics for CRC.

5.4.3 Scanner Variability

For the validation of the model, consecutive cuts are acquired. After the staining procedure, the images are digitized using two scanners, A and B, from the same constructor. Still, they present different characteristics that can hinder segmentation model performances [80]. Firstly, scanner A uses an old, low-tier camera, while scanner B relies on

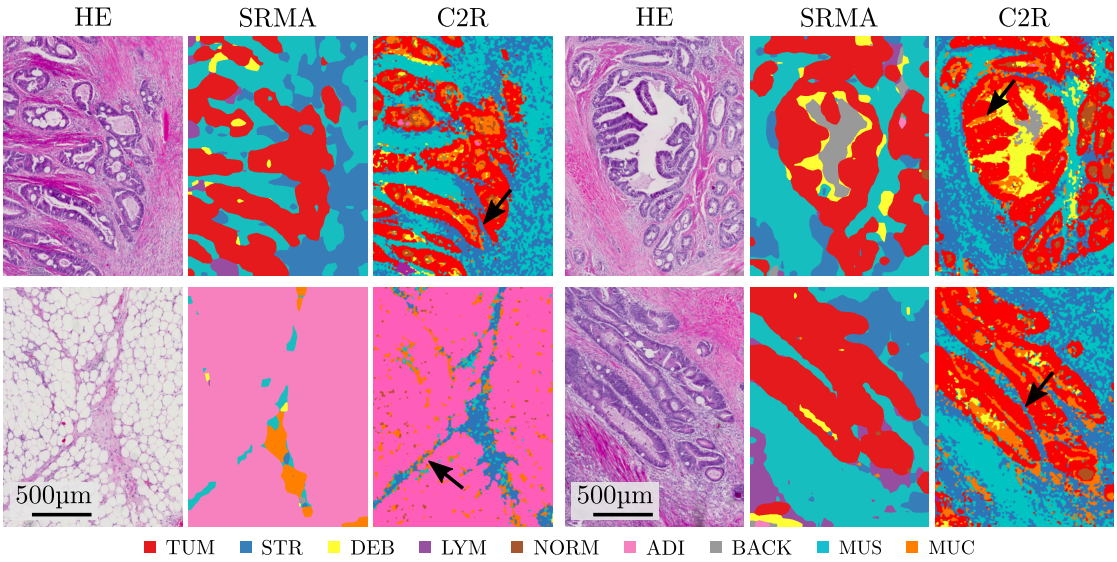
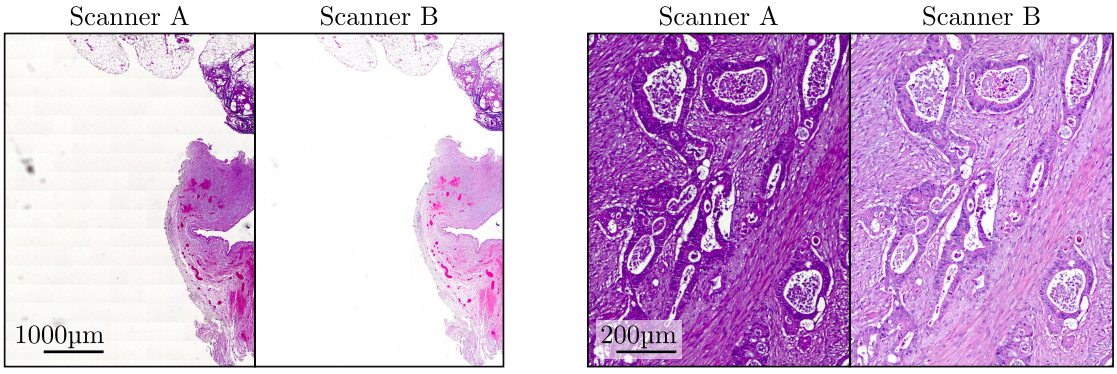


Figure 5.7 – Comparison of WSI local segmentation between the proposed approach and previous SRMA [2] work. The arrows highlight specific areas where C2R is able to achieve detailed segmentation.



(a) Low magnification (b) High magnification

Figure 5.8 – Scanners visual comparison at (a) low and (b) high magnification. We display the same regions acquired with two scanners, A and B.

a recent, high-quality camera. Secondly, scanner B uses gamma calibration, while the feature is deactivated in scanner A. This disparity allows us to compare the model's performance under different acquisition settings. In Figure 5.8, we display examples of the two scanned images at low and high magnification.

The difference in color calibration is visible between the two images. For scanner B, the images appear brighter, and the contrast between the hematoxylin and eosin stains is more visible. In addition, between the two acquisitions, slides were manually cleaned. On the images from scanner A, we identify dust on the glass that causes obstructions and focusing issues for the camera.

Table 5.2 – Comparison of the performance of C2R under scanner variation in $\mathcal{D}_{\text{C2R-ROI}}$. Results are averaged over 10 runs. We use an unpaired t-test with respect to the top result. The single level setting use only last ViT layer and multilevel use layers $\{3, 6, 9, 12\}$. We report class-wise F_1 score and DCD.

		Tumor		Stroma		All
Methods	Scanner	DCD [150]	F_1	DCD [150]	F_1	Macro-F_1
Baseline						
SRMA[2]	A	0.129 ± 0.002	65.9 ± 0.7	0.101 ± 0.011	63.3 ± 4.2	64.6 ± 2.1
Single level						
C	A	0.103 ± 0.016	65.6 ± 1.8	0.039 ± 0.007	66.1 ± 5.3	65.9 ± 3.2
K	A	0.075 ± 0.008	70.5 ± 1.0	0.044 ± 0.003	61.7 ± 2.4	66.1 ± 1.4
CK	A	0.089 ± 0.011	67.8 ± 2.0	0.036 ± 0.004	68.9 ± 3.6	68.4 ± 2.5
Multilevel						
ML-C	A	0.101 ± 0.022	69.5 ± 3.4	0.031 ± 0.004	75.5 ± 3.9	72.5 ± 1.9
ML-K	A	0.075 ± 0.021	72.4 ± 3.2	0.041 ± 0.008	67.0 ± 6.7	69.7 ± 4.6
ML-CK	A	0.086 ± 0.017	72.3 ± 2.3	**0.031** ± 0.008	75.3 ± 5.8	73.8 ± 2.6
Baseline						
SRMA[2]	B	0.108 ± 0.006	68.1 ± 2.5	0.084 ± 0.006	70.1 ± 2.3	69.1 ± 1.2
Single level						
C	B	0.071 ± 0.010	70.4 ± 1.4	**0.031** ± 0.005	71.7 ± 4.8	71.1 ± 2.7
K	B	0.048 ± 0.003	72.2 ± 0.7	0.038 ± 0.003	66.1 ± 2.6	69.2 ± 1.4
CK	B	0.058 ± 0.007	71.7 ± 1.7	**0.029** ± 0.003	73.9 ± 3.8	72.8 ± 2.3
Multilevel						
ML-C	B	0.056 ± 0.010	**75.2** ± 3.1	**0.026** ± 0.007	**80.6** ± 6.1	**77.8** ± 4.1
ML-K	B	**0.040** ± 0.006	**75.0** ± 2.1	0.038 ± 0.003	70.9 ± 4.8	72.9 ± 3.3
ML-CK	B	0.050 ± 0.011	**76.5** ± 2.7	**0.026** ± 0.004	**79.4** ± 5.5	**78.0** ± 3.2

Abbreviations: $\mathcal{L}_{\text{cls}}$ (C), $\mathcal{L}_{\text{knn}}$ (K), Multilevel (ML).

In Table 5.2, we compare the performance of the proposed architecture on the different scanners. We report the F_1 score and density-aware Chamfer distance (DCD) for tumor and stroma, as well as the macro-F_1 over both classes. The use of the DCD is a good indicator of the granularity of the prediction. We observe a significantly higher error rate when computing DCD for the tumor on scanner A. As the model tends to miss small tumor areas due to the lack of color contrast, the distance to the closest set tends to increase and thus DCD with it. This phenomenon is less visible on stroma as the class is usually composed of large and dense areas. The evaluation of the F_1 score also highlights the difference between the two scanners. Note that during the training procedure, the augmentation includes transformations that take into account gamma correction and color distortion. However, such transformations are insufficient when the color space gap is too large. A visualization of tissue segmentation applied to tumor and stroma for scanner comparison is available in appendix section D.1.

Table 5.3 – Ablation studies of C2R architecture on SemiCol challenge. Results are averaged over 10 runs. We use an unpaired t-test with respect to the top result. We report class-wise F_1 score and challenge specific loss SemiCol-F_1.

	F_1 score								SemiCol-F_1
Methods	ADV	LYM	MUC	NORM	MUS	DEB	TUM	CSTR	All
Single level									
C	36.1	42.7	43.2	25.9	48.4	24.3	47.1	28.2	34.3 ± 0.8
K	36.0	31.3	37.6	30.0	46.9	22.1	45.5	22.9	31.8 ± 0.7
CK	36.3	43.3	41.0	25.4	49.2	24.2	47.4	27.5	34.2 ± 0.7
Multilevel									
ML-C	38.9	61.9	52.2	31.8	56.3	35.9	57.9	29.1	42.2 ± 1.7
ML-K	38.1	43.6	40.6	35.1	55.1	30.3	49.0	22.9	36.4 ± 2.2
ML-CK	41.8	58.1	50.1	35.6	57.8	36.0	58.6	29.5	42.6 ± 1.6
Visual consistency									
ML-C-P	42.1	61.8	53.4	35.4	55.9	35.0	59.5	**31.2**	43.4 ± 1.7
ML-C-R_{L1}	**46.5**	**70.6**	58.6	44.0	64.3	**45.9**	60.4	**32.8**	48.3 ± 1.8
ML-C-R_{H}	**45.0**	58.3	54.0	37.6	60.3	35.9	57.8	30.6	43.7 ± 1.5
ML-C-PR_{L1}	**45.3**	**67.8**	59.2	**46.8**	61.6	**43.4**	61.8	**33.2**	48.1 ± 1.9
ML-C-PR_{H}	**46.0**	61.7	54.3	39.3	60.0	36.6	58.5	**31.5**	44.6 ± 0.9
Self-correlation									
ML-C-$S_{0.05}$	39.8	**68.9**	60.7	41.4	62.1	37.7	61.9	**31.6**	46.6 ± 1.7
ML-C-$S_{0.20}$	40.9	**71.4**	54.5	31.4	61.3	37.1	61.9	30.4	45.1 ± 0.8
ML-C-S_{MVA}	38.3	**72.9**	56.0	39.0	60.1	39.3	61.7	30.6	46.0 ± 1.3
Visual & Self-correlation									
ML-C-P-$S_{0.05}$	39.3	**71.5**	60.9	41.2	61.9	38.2	**63.0**	**33.1**	47.2 ± 2.0
ML-C-PR_{L1}-$S_{0.05}$	42.5	**75.8**	**71.4**	**47.6**	**64.7**	**46.0**	**64.9**	**32.8**	**51.1** ± 2.9
ML-C-PR_{H}-$S_{0.05}$	40.5	**73.9**	61.7	**45.5**	62.8	**41.9**	**62.4**	**31.9**	48.3 ± 2.3

Abbreviations: $\mathcal{L}_{\text{cls}}$ (C), $\mathcal{L}_{\text{knn}}$ (K), $\mathcal{L}_{\text{cls}}^{\text{photo}}$ (P), $\mathcal{L}_{\text{reg}}^{\text{photo}}$ (R), $\mathcal{L}_{\text{seg}}$ (S), Multilevel (ML), l^1-norm (L1), histogram matching (H).

5.4.4 Ablation Study - SemiCol Challenge

To further validate our approach, we use data from the SemiCol online challenge[1]. The challenge includes two tasks: classification of WSIs and segmentation of CRC tissue.

For the first experiment, we apply our model trained on K19 to the challenge data. As the class definitions from the challenge differ from the K19 data, we assume complex stroma (CSTR) is linked to STR and that advent (ADV) is correlated to ADI. In addition, we discard blood (BLOOD) class as it is absent from the source data. The results are reported in Table 5.3. We report F_1 score for each class as well as SemiCol macro-F_1. The metric is defined as the standard macro-F_1 but with twice the importance given to TUM class.

Overall, the use of both visual consistency and self-correlation improves the performance of the model. We observe a substantial benefit in using the multilevel aggregation

[1] `https://www.semicol.org/` last accessed on 26/05/23.

Table 5.4 – Evaluation of C2R architecture when trained on SemiCol challenge data. Results are averaged over 10 run. We use an unpaired t-test with respect to the top result. We report class-wise F_1 score and challenge specific loss SemiCol-F_1.

	F_1 score									SemiCol-F_1
Methods	ADV	BLOOD	LYM	MUC	NORM	MUS	DEB	TUM	CSTR	All
Trained on K19										
ML-C	38.9	0.0	61.9	52.2	31.8	56.3	35.9	57.9	29.1	42.2 ± 1.7
ML-C-P	42.1	0.0	61.8	53.4	35.4	55.9	35.0	59.5	31.2	43.4 ± 1.7
ML-C-PR_{L1}	45.3	0.0	67.8	59.2	46.8	61.6	43.4	61.8	33.2	48.1 ± 1.9
ML-C-PR_{H}	46.0	0.0	61.7	54.3	39.3	60.0	36.6	58.5	31.5	44.6 ± 0.9
ML-C-$S_{0.05}$	39.8	0.0	68.9	60.7	41.4	62.1	37.7	61.9	31.6	46.6 ± 1.7
ML-C-PR_{L1}-$S_{0.05}$	42.5	0.0	75.8	71.4	47.6	64.7	46.0	64.9	32.8	51.1 ± 2.9
ML-C-PR_{H}-$S_{0.05}$	40.5	0.0	73.9	61.7	45.5	62.8	41.9	62.4	31.9	48.3 ± 2.3
Trained on SemiCol										
ML-C	72.7	59.3	62.5	62.8	55.6	80.1	62.5	59.5	38.0	61.3 ± 2.4
ML-C-P	73.8	59.2	61.4	61.8	56.8	80.8	62.2	63.5	43.9	62.7 ± 2.8
ML-C-PR_{L1}	75.1	65.9	71.3	72.1	70.8	85.6	69.6	68.0	**47.7**	69.4 ± 2.0
ML-C-PR_{H}	71.8	57.7	63.8	59.9	57.4	78.8	62.8	63.3	41.7	62.0 ± 3.1
ML-C-$S_{0.05}$	**78.9**	**73.3**	**80.4**	**77.5**	**82.2**	**86.8**	**76.5**	**71.0**	**47.6**	74.5 ± 1.7
ML-C-PR_{L1}-$S_{0.05}$	**79.5**	**74.2**	**82.6**	**78.4**	**83.5**	**87.4**	**78.9**	**73.5**	**51.6**	**76.3** ± 2.1
ML-C-PR_{H}-$S_{0.05}$	**79.3**	**71.6**	**81.7**	**78.8**	**81.7**	**87.0**	**78.1**	**72.4**	**50.5**	**75.3** ± 2.4

Abbreviations: $\mathcal{L}_{\text{cls}}$ (C), $\mathcal{L}_{\text{knn}}$ (K), $\mathcal{L}_{\text{cls}}^{\text{photo}}$ (P), $\mathcal{L}_{\text{reg}}^{\text{photo}}$ (R), $\mathcal{L}_{\text{seg}}$ (S), Multilevel (ML), l^1-norm (L1), histogram matching (H).

compared to the single-level setting as in the in-house validation set. In addition, using the l^1-norm for visual consistency shows better results. The difference is particularly notable for the detection of tumor and mucin tissue. A surprising outcome is the poor performance of the self-correlation loss on the ADV term. Here, the model predicts the adipose central areas as background tissue as they appear empty. This behavior is expected as the self-correlation ensures that regions with high feature similarity share the same label. Moreover, using l^1 regularization for visual consistency significantly impacts muscle detection, even outperforming other proposed loss compositions.

Adaptation to Challenge Data

To prove the robustness of our model to weakly-labeled data, we retrain our model using the SemiCol challenge data. We create weak labels from the provided segmentation annotation. For each annotated image, we apply majority voting across the whole region. Here, we aim to mimic a lazy annotator that would give single labels to large areas without caring about precise annotation. Such an approach allows the annotator to efficiently label WSIs in a small amount of time. The training set contains 1,759 tiles of 3000px × 3000px from 20 different WSI. Out of this set, we extracted 21,783 annotated tiles of size 500px × 500px. The classes are distributed as follow: 6,971 background, 277 blood, 197 lymphocytes, 2,604 normal mucosa, 420 mucin, 3,111 muscle, 387 debris, 5,549 advent, 1359 tumor and 889 complex stroma examples.

The generated data are largely imbalanced with a ratio of $N_{\text{BACK}}/N_{\text{LYM}} = 35$ between the highest and least populated classes. Imbalanced data are expected in clinical applications as the tissue classes are scarcely equally distributed. When training the architecture, we pay attention to increasing the sampling rate of the underrepresented classes to ensure proper representation of all tissues. We present the results in Table 5.4, where we report previous results (*i.e.* trained based on K19) and adapted results (*i.e.* trained with SemiCol weakly-labeled data).

We observe a significant improvement in the prediction performance over all classes. The detection of NORM, MUS, and DEB increases by over 20%. The base architecture with the multilevel setting outperforms all models trained using K19 data. Moreover, we observe that with fewer data, the main contribution comes from the self-correlation loss. The term helps the model to create homogeneous class areas. The benefit of visual consistency is only visible for the overall segmentation.

5.5 Conclusion

In this chapter, we present an approach for tissue segmentation, where we take advantage of coarsely-labeled data. We first build a shallow model based on publicly available data. Then, we use the model to process our in-house cohort and extract pseudo labels from more than 600 WSIs. Next, we use pseudo labels as input for our C2R algorithm. As the inputs are extracted from WSIs, we ensure that the data represent heterogeneous classes and complex tissue interaction. Here, we prove we can benefit from weakly-labeled public data to achieve fine-grained segmentation. More importantly, the proposed architecture does not require additional annotation to be trained, thus saving experts precious time.

To validate the performance of our segmentation, we take advantage of consecutive cuts stained with HE, IHC, and Trichrome to generate segmentation labels for tumor and stroma classes automatically. By doing so, we again avoid the need for additional annotations. Moreover, the acquisition of consecutive cuts is reasonably cheap, which makes it an affordable option for clinical institutes. In addition, as the model is based on a coarse to refined logic, designing new segmentation tasks would not demand extensive manual annotation. The coarse labeling of WSIs would be enough to achieve precise prediction. This claim is supported by our experiment on the SemiCol challenge data, where only weakly-labeled areas are available.

With the proposed approach, we can predict fined-grained segmentation maps given weakly-labeled data. However, solely predicting tissue outputs is pointless if not linked to a practical end task. In the next chapter, we take advantage of our model to predict clinically relevant metrics automatically.

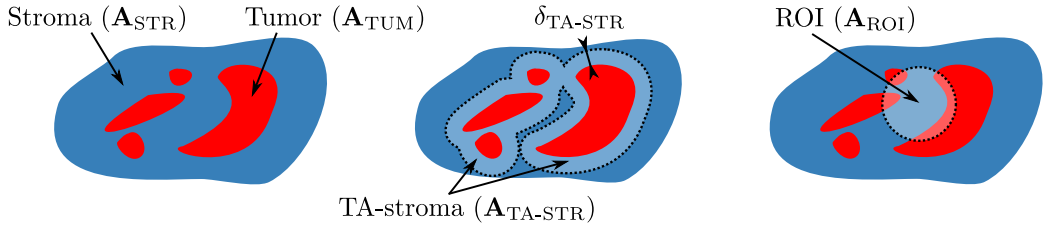
6 Building Clinically Relevant Metrics

The previous chapters focused on the creation and optimization of various self-supervised learning architectures. These frameworks can solve complicated tasks, learn from open source weakly-labeled data, generate representations of complex tissue structure, account for whole slide image (WSI) domain shift, or predict fine-grained segmentation maps.

However, the presented applications and experiments are limited to classifying and segmenting well-known public datasets that lack clinical motivations. This chapter uses the previously designed architecture to build clinical metrics and perform survival analysis to connect our research to medical applications. Clinical metric assessment often relies on pathologists' visual assessment, which might come with a few limitations. The first is that visual annotation tends to have low interobserver agreement (IOA), which can lower the confidence of survival model scores. The second is that manual annotation of clinical cohorts, including hundreds of patients and slides, is tedious. Finally, the third is that the need for manual annotation makes the generalization of the method to other cohorts tricky.

In this work, we propose using our tissue segmentation model to build a fully automated pipeline to predict clinically relevant metrics on various cohorts without the need for external annotation [3]. Using an automated approach, we aim to solve the issues on IOA, manual annotations, and generalization. We first introduce multiple clinical metrics as tumor to stroma ratio (TSR) and tumor border configuration (TBC) that are known to have high predictive value in colorectal cancer (CRC) in section 6.1 and 6.2, respectively. Then, we further explore other less well-established clinical features such as tumor to mucin ratio (TMR) and tissue distribution in section 6.3. Next, we compare our results with expert annotation on different clinical cohorts, compute correlation with other reported clinical variables, and perform univariate and multivariate survival analysis in section 6.4. Finally, we conclude our chapter in section 6.5. To further contribute to open research, we make our code available on GitHub[1].

[1] `https://github.com/christianabbet/WSImetrics_CRC`.



(a) Detection (b)Associated stroma (c) Region of interest

Figure 6.1 – Identification of main tissue region for TSR estimation. (a) Detection of tumor (A_{TUM}) and stroma (A_{STR}) tissue from a WSI. (b) Tumor-associated stroma ($\text{A}_{\text{TA-STR}}$) in range $\delta_{\text{TA-STR}}$ from main tumor. (c) Region of interest A_{ROI}.

6.1 Tumor to Stroma Ratio

Tumor to stroma ratio has been shown to be an independent prognostic factor in CRC [3, 131, 154, 133]. The metrics aim to quantify the interaction between tumor tissue and the surrounding stromal content. Dense tumor areas are more likely to be linked with less aggressive tumor progression and, thus, better survival prognosis. On the contrary, regions with sparse tumor areas are more likely to have high tumor budding, more invasive patterns, and, therefore, poor prognosis factors [141].

Recent works attempt to predict TSR in a (semi-)automated fashion. In [52], they rely on pixel-wise annotation to detect tumor and stroma components. In [154], they use superpixel segmentation and feature extraction to predict TSR in tissue microarray (TMA) for breast cancer. The approach relies on fine-grained annotations by expert pathologists, which are tedious to acquire. In [159], they generate a coarse segmentation map based on patch labels to estimate TSR. Here, TMAs are assumed to be representative of the main WSI and thus require manual annotations. In [106], they propose quantifying the desmoplastic reaction at the tumor border. However, training the segmentation architecture and identifying the tumor front depends, again, on human inputs.

In Figure 6.1, we highlight the main components needed to compute TSR. Given a hematoxylin and eosin (HE) WSI, we assume a segmentation is generated where classes tumor (TUM) and stroma (STR) are available. The tumor $\mathbf{A}_{\text{TUM}}$ and stroma $\mathbf{A}_{\text{STR}}$ areas are defined as the area that is segmented by the model as tumor and stroma, respectively. In addition, we introduce the tumor-associated stroma (TA-STR) area $\mathbf{A}_{\text{TA-STR}}$ that represent the neighborhood of the tumor. It is estimated as the WSI stroma tissues that are in the range $\delta_{\text{TA-STR}}$ of the main tumor [106].

Given the introduced components, we define TSR at the WSI level:

$$\text{TSR}_{\text{WSI}} = \frac{|\mathbf{A}_{\text{TUM}}|}{|\mathbf{A}_{\text{TUM}} \cup \mathbf{A}_{\text{STR}}|}. \tag{6.1}$$

When computing TSR at the WSI level, we might include stroma content from normal areas. This can bias the metric by over-representing STR and thus underestimating TSR. This is even more problematic as low TSR has been proven to be correlated to worse survival probability. To tackle this issue, we replace the estimation of the STR with TA-STR such that the estimation of TSR directly depends on stroma content surrounding the tumor:

$$\mathrm{TSR}_{\mathrm{TA}} = \frac{|\mathbf{A}_{\mathrm{TUM}}|}{|\mathbf{A}_{\mathrm{TUM}} \cup \mathbf{A}_{\mathrm{TA\text{-}STR}}|}. \tag{6.2}$$

Finally, we define a TSR estimation based on a region of interest (ROI). We assume we are able to identify a localized region $\mathbf{A}_{\mathrm{ROI}}$ that we believe is a good representation of the overall tumor progression. Based on the area, we update the TSR estimation:

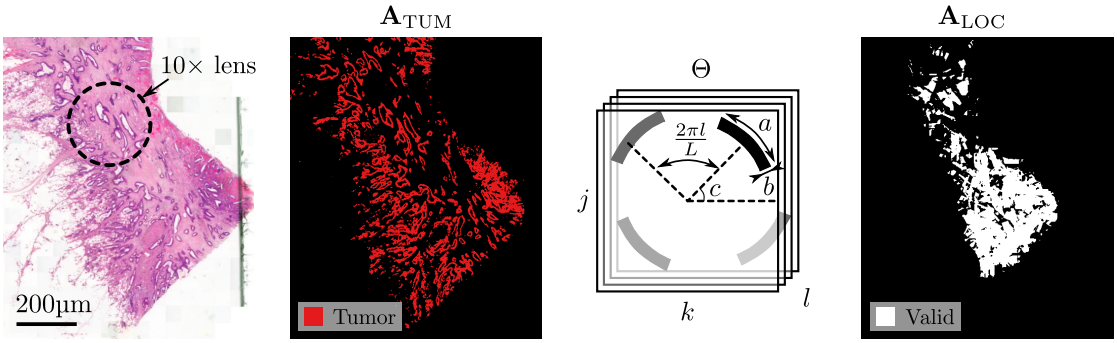
$$\mathrm{TSR}_{\mathrm{ROI}} = \frac{|\mathbf{A}_{\mathrm{TUM}} \cap \mathbf{A}_{\mathrm{ROI}}|}{|(\mathbf{A}_{\mathrm{TUM}} \cup \mathbf{A}_{\mathrm{STR}}) \cap \mathbf{A}_{\mathrm{ROI}}|}. \tag{6.3}$$

The definition and identification of the ROI is based on clinical recommendations and is discussed in the next section.

6.1.1 Region of Interest Identification

Currently, TSR is not reported in routine diagnostics, as there are no binding guidelines. However, there exists a scoring recommendation [141]. On slides from the most invasive tumor part, the area with the highest amount of stroma and where tumor cells are present in all "four directions" of the image field is selected using a 10× lens. Then, the amount of tumor and stroma in the ROI are estimated and scored as 10% increments. As a result, the estimation of TSR for a patient is given by a single ROI that matches the previously mentioned criteria. To solve this task, we use a two-step procedure. First, we explain how we identify potential region candidates (*i.e.* regions where tumor cells are present in all "four directions") and then detail how we select the final ROI from all matching candidates.

We define the concept of "four directions" as the region where the tumor is homogeneously distributed around the ROI central point. If we draw L equally spaced lines (inter-angle of $2\pi/L$) that radiate from the ROI center, each line should encounter a tumor area in a close range. We formalize this approach by defining a detection area $\mathbf{A} \in [0,1]^{W \times H \times C}$ that represents the result of a segmentation algorithm where W, H are the dimensions of the area and C the number of detected classes. We denote as $\mathbf{A}_{\mathrm{STR}}, \mathbf{A}_{\mathrm{TUM}} \in [0,1]^{W \times H}$ the channels of $\mathbf{A}$ that contain the stroma and tumor class detection probability, respectively. To detect potential ROI candidates with tumor presence, we define a filter



(a) Input WSI (b) Segmentation (c) Detection filters (d) Valid locations

Figure 6.2 – Estimation of valid tumor regions in all directions. (a) Input WSI with 10× lens reference. (b) Tumor segmentation map. (c) Detection sub-filter Θ with hyperparameters L, a, b, c. (d) Post filtering and localization of valid ROI center where the tumor is present in all directions.

$\Theta = \begin{pmatrix}\Theta_1 & \dots & \Theta_L\end{pmatrix} = (\theta_{jkl})_{1\leq j,k\leq N\,,1<\leq l\leq L}$. The filter comprises L sub-filters of shape $N \times N$ that form circle sections. The size N is set such that the filter size matches with the 10× lens dimensions. The filter is defined as:

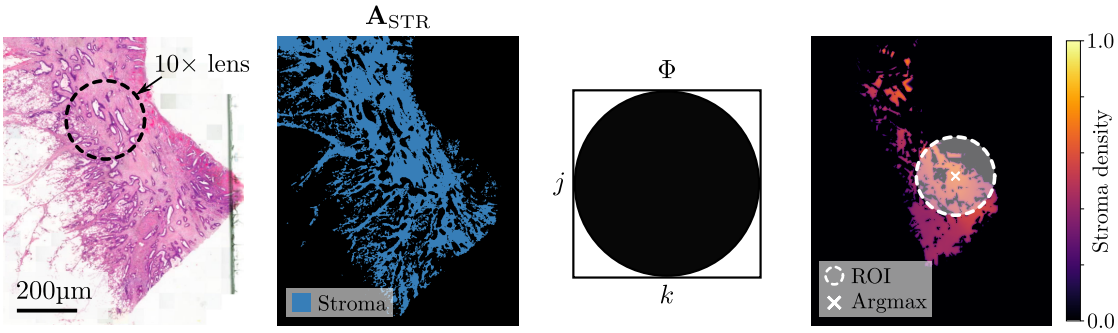
$$\theta_{jkl}(a,b,c) = \Pi\left(\frac{1}{2} + \frac{r - N/2}{a}\right)\Pi\left(\frac{\phi}{b}\right), \tag{6.4}$$

$$\text{with} \quad z = \left(j - \frac{N}{2}\right) + i\left(k - \frac{N}{2}\right), \quad r = |z|, \quad \text{and} \quad \phi = \arg\left(z\, e^{i(\frac{2\pi l}{L}+c)}\right),$$

where i is the imaginary component, Π the rectangular function, j, k, l the filter indexes, and arg the complex number argument (*i.e.* angle with respect to positive real axis). An overview of the created filter is given in Figure 6.2. The hyperparameters a, b, and c control the filter width, depth, and offset angle, respectively. Each sub-filter is separated by an angle of $\frac{2\pi}{L}$. We apply sub-filters on the detected tumor area and use a threshold δ_{ROI} to check for ROI centers candidates:

$$\mathbf{A}_{\text{LOC}} = \max\left(\left(\sum_{l=1}^{L}\frac{1}{|\Theta_l|}(\mathbf{A}_{\text{TUM}} * \Theta_l) \geq \delta_{\text{ROI}}\right) - L + 1, 0\right), \tag{6.5}$$

where the max function is applied to all entries of the matrix individually. We limit the convolution operator to the dimensions of $\mathbf{A}_{\text{TUM}}$. With the presented formula, an element of $\mathbf{A}_{\text{LOC}}$ is considered as valid ROI center location if the convolution between the tumor area and the sub-filters exceeds the threshold value δ_{ROI} for all L channels. Increasing the value of L generates more sub-filters and thus further enforces the presence of the tumor around the patch center.



(a) Input WSI (b) Segmentation (c) Averaging filter (d) ROI location

Figure 6.3 – Localization of high stroma concentration for TSR estimation. (a) Input WSI with high stroma content. (b) Segmented stroma area. (c) Average filter as a circular disk. (d) Local stroma density and identification of max value across WSI.

With $\mathbf{A}_{\text{LOC}}$, we get a list of ROI potential locations. Out of localized candidates, we want to select the area with the highest stroma content to compute TSR. To do so, we use the same logic as before and build a second filter that computes the local average stroma content across the map. The filter $\Phi = (\phi_{jk})_{1 \leq j,k \leq N}$ is defined as:

$$\phi_{jk} = \Pi\left(\frac{r}{N} - \frac{1}{2}\right) \qquad \text{and} \qquad r = \sqrt{\left(j - \frac{N}{2}\right)^2 + \left(k - \frac{N}{2}\right)^2}. \tag{6.6}$$

The filter represents a disk of diameter N. The procedure is depicted in Figure 6.3. When convoluted with an area, it averages the local predictions. The filter is applied to the stroma detection and masked using the previously computed ROI valid locations. Out of the valid locations, the selected region $\mathbf{A}_{\text{ROI}} = (a_{j,k})_{1 \leq j \leq W, 1 \leq k \leq H}$ is given as the circular area with radius $\frac{N}{2}$ centered in the point with highest stroma content:

$$a_{j,k} = \mathbb{1}_{\text{dist}((j,k),(l,m)) < N}, \tag{6.7}$$

$$\text{and} \quad \underset{l \in \{1,\dots W\}, m \in \{1,\dots H\}}{\arg\max} \left((\mathbf{A}_{\text{STR}} * \Phi) \cap \mathbf{A}_{\text{LOC}}\right)_{l,m},$$

where $\text{dist}(\cdot,\cdot)$ computes the euclidean distance between two points coordinates and l, m are the coordinated of the ROI center. The TSR is then estimated using Equation 6.3.

$$\text{TSR}_{\text{ROI}} = \frac{|\mathbf{A}_{\text{TUM}} \cap \mathbf{A}_{\text{ROI}}|}{|(\mathbf{A}_{\text{TUM}} \cup \mathbf{A}_{\text{STR}}) \cap \mathbf{A}_{\text{ROI}}|}. \tag{6.3 recall}$$

In practice, we apply Gaussian blurring on top of the detected stroma density to

merge potential local maxima and get a smoothed ROI localization. Moreover, we sometimes have access to a handful of WSIs per patient. When looking at the scoring recommendations, we should select the case with the highest stroma content, hence the lowest computed TSR across slides. However, it assumes we can access all patients' slides, which is scarcely true. If only a few slides are available per patient, we recommend processing TSR for individual WSI and then averaging the values across slides.

With the presented approach, the estimation of the TSR is based on the evaluation of a single ROI. As a result, the algorithm might be sensitive to local maximums and minimums. Moreover, as the method explicitly looks for the region with high stormal content, it might oversample regions with low TSR and bias survival predictions. To reduce the impact of outliers, we propose to investigate the use of top-K ROI detections to estimate the TSR at the WSI level. To do so, we start by predicting the first ROI as before. Then, we remove the former ROI detection from the prediction map and compute the second ROI. The procedure is repeated until we get all K estimates. The top K predictions are then averaged to produce the final estimate:

$$\text{TSR}_{\overline{\text{ROI}}} = \frac{1}{K} \sum_{i=1}^{K} \text{TSR}_{\text{ROI}i} \,, \tag{6.8}$$

where $\text{TSR}_{\text{ROI}i}$ is the metric estimation at step i.

6.2 Tumor Border Configuration

The TBC is defined as the configuration of the tumor invasive front [70]. The margin can be classified as either expanding/pushing (*i.e.* "tumor reasonably well circumscribed") or infiltrating (*i.e.* "tumor invading in a diffuse manner with widespread penetration of normal tissues"). An illustration of the TBC is presented in Figure 6.4. The TBC is evaluated at the WSI level. A WSI that shows a fully infiltrating tumor pattern is given a score of 0. On the contrary, a WSI with a solely pushing pattern is scored with 1. All intermediary configurations lie in the range TBC $\in [0, 1]$.

Multiple works focus on the estimation and use of TBC in CRC [84, 168, 117]. In [168], the inclusion of TBC in survival analysis shows better stratification of stage II patients. In [117], the work is extended to stage III patients to show similar results. Both studies rely on manually annotated images from single cohorts.

In this section, we propose a fully automated way to extract TBC from WSI predictions. To do so, we first explain how we find the delimitation between normal and tumor tissue to detect tumor border (TB) in subsection 6.2.1. As a second step, we present three different approaches to estimate TBC using normal vectors (subsection 6.2.2), tumor

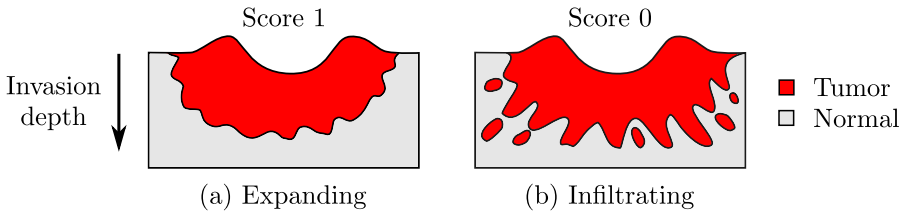


Figure 6.4 – Visualization of tumor border configuration (TBC). (a) Expanding border: pushing tissue. (b) Infiltrating: invasion of normal tissue in a diffuse manner. Figure adapted from [70].

ratio (subsection 6.2.3), and tumor interaction (subsection 6.2.4) based on TB.

6.2.1 Tumor Border

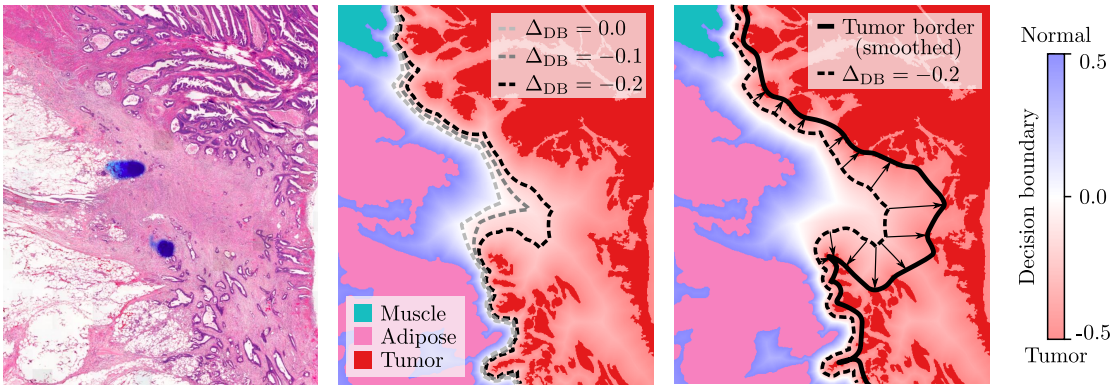
The definition of the tumor area slightly differs from the TSR section. For TSR estimation, we use raw tumor detection from segmentation models as the main tumor area. However, when estimating TB, we must consider the attached necrotic and mucinous tissues to identify all tumor-related components. In this section, we define the tumor area as the combination of the tumor-detected tissue and its surrounding debris and mucin. We use a region-growing approach to expand the tumor area to neighboring tissues. For more information about the creation of the tumor area, please refer to the additional content in section E.1.

To perform the identification of the TB, we assume the tumor progresses linearly from the inner colon (normal mucosa) to the muscle area to reach the outer fat tissue (adipose) finally. As a result, we can look for the decision boundary that separates the tumor tissue (*i.e.* tumor, debris, and mucin) from the normal tissues (*i.e.* normal mucosa, muscle, and adipose). For notation simplicity, we define as $\mathbf{p} \in \mathbf{A}$ a point that belongs to the prediction map $\mathbf{A}$ (*i.e.* the coordinates of the point are part of the area). We define the distance of a point $\mathbf{p}$ to the nearest normal and tumor tissue:

$$d_{\text{TUM}}(\mathbf{p}) = \min_{\mathbf{q} \in \mathbf{A}_{\text{TUM}}} \|\mathbf{p} - \mathbf{q}\|_2 \qquad \text{and} \qquad d_{\text{NORM}}(\mathbf{p}) = \min_{\mathbf{q} \in \mathbf{A}_{\text{NORM}}} \|\mathbf{p} - \mathbf{q}\|_2, \tag{6.9}$$

where $\mathbf{A}_{\text{TUM}}$ and $\mathbf{A}_{\text{NORM}}$ are defined as tumor and normal tissue areas, respectively. We then define the decision boundary as the function:

$$\Delta_{\text{DB}}(\mathbf{p}) = \frac{d_{\text{TUM}}(\mathbf{p}) - d_{\text{NORM}}(\mathbf{p})}{2(d_{\text{TUM}}(\mathbf{p}) + d_{\text{NORM}}(\mathbf{p}))}. \tag{6.10}$$



(a) Input WSI (b) Decision boundaries (c) Fit to tumor

Figure 6.5 – Estimation of TB from HE image. (a) Input image from HE scan. (b) Identification of the normal areas (norma mucosa/muscle/adipose) and tumor (tumor/debris/mucin). We compute the distance to the nearest tissue and show decision boundaries function Δ_{DB} based on different thresholds. (c) Regression of decision boundary toward tumor tissue and smoothing to obtain TB estimation.

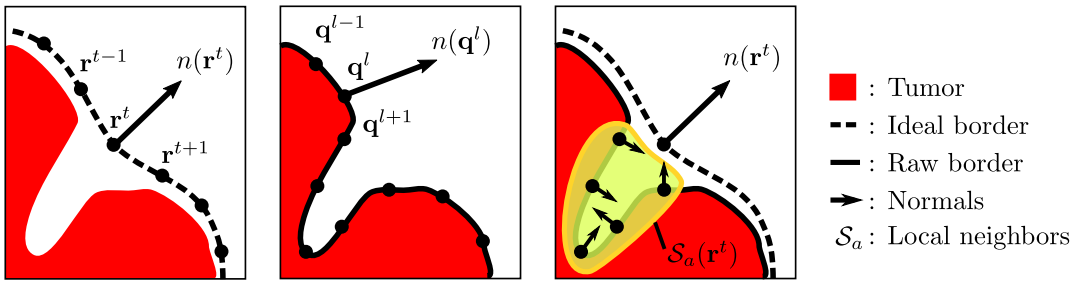
The function is designed such that:

$$\lim_{d_{NORM} \to 0} \Delta_{DB} = \frac{1}{2} \quad \text{and} \quad \lim_{d_{TUM} \to 0} \Delta_{DB} = -\frac{1}{2}. \tag{6.11}$$

The decision boundary is contained in the interval $[-\frac{1}{2}, \frac{1}{2}]$ and equal to 0 when the distance to both region is equal. Values of $-\frac{1}{2}$ and $\frac{1}{2}$ mean direct contact with tumor and normal tissue, respectively. The estimation of the TB is visualized in Figure 6.5. To estimate the TB, we first follow the contour of the decision function given a fixed value. The selection of the threshold is performed empirically such that it creates a coherent contour that outlines the shape of the tumor. When looking at the example in Figure 6.5b, we observe a gap between the boundary and the main tumor blob. Ideally, the estimated TB should tightly follow the outline of the tumor.

To reduce the gap, we iteratively regress the decision line toward the tumor area. For each point of the border, we move it to the nearest tumor point. The process is repeated multiple times until convergence. The resulting line now directly fits the main tumor outline. Finally, we use spline [45] fitting to smooth the representation and get our TB estimation. When fitting the spline functions, we uniformly sample the tumor border to ensure fair representation of each border point.

For the rest of the chapter, we define the set of N points $\{\mathbf{r}^t \in \mathbf{A}\}_{t=1}^{N}$ that represents our estimated TB points. We assume the points are ordered along the TB such that the contour normals point outward from the tumor area.



(a) Ideal border (b) Raw border (c) Dot product

Figure 6.6 – Computation of TBC based on local border normals. We use the ideal tumor front as a pushing reference and compare it with the raw tumor outline. (a) Estimated ideally pushing tumor border points $\mathbf{r}^t$ with normals $n(\mathbf{r}^t)$. (b) Raw tumor outline points $\mathbf{p}^l$ with normals $n(\mathbf{p}^l)$. (c) We perform the dot product between elements of the set $\mathcal{S}_a$ and $\mathbf{r}^t$ to estimate local TBC.

6.2.2 Normal Product

This section presents our first approach to compute TBC. To do so, we assume the TB is a good estimation of an ideally pushing border as it harmoniously follows the tumor's outline. As a result, any shape that differs from that reference line would be considered infiltrating. An illustration of the procedure is depicted in Figure 6.6, where we can see the set of point $\mathbf{r}^t$ that represents the TB. In addition, we introduce the set of points $\{\mathbf{q}^l \in \mathbf{A}\}_{l=1}^{M}$ that represent the "raw" border of the tumor map $\mathbf{A}_{\text{TUM}}$.

So far, we have two sets of points that do not align and represent different borders. We aim to measure how much the raw points $\mathbf{q}^l$ differ from the ideally pushing line formed by points $\mathbf{r}^t$. To do so, we first compute the normal vector of points $\mathbf{r}^t$:

$$n(\mathbf{r}^t) = \frac{1}{\|\mathbf{r}^{t+1} - \mathbf{r}^{t-1}\|_2} \begin{pmatrix} (r_y)^{t-1} - (r_y)^{t+1} \\ (r_x)^{t+1} - (r_x)^{t-1} \end{pmatrix}, \tag{6.12}$$

where $(r_x)^t$ and $(r_y)^t$ are the x and y coordinate of point $\mathbf{r}^t$. As previously mentioned, we assume the points are ordered along the border and that their normals always head outward from the tumor area. We do the same with the points from the raw borders and get $n(\mathbf{q}^l)$. Next, we create the neighborhood of TB points:

$$\mathcal{S}_a(\mathbf{r^t}) = \{l \mid \|\mathbf{q}^l - \mathbf{r}^t\|_2 \leq \|\mathbf{q}^l - \mathbf{r}^k\|_2 \, \forall k \in \{1, \ldots, N\}\} \quad l \in \{1, \ldots, M\}. \tag{6.13}$$

Here, the set $\mathcal{S}_a(\mathbf{r^t})$ contains the indexes of the raw tumor points that have $\mathbf{r^t}$ as closest element. The local TBC estimation as index t is given as the dot product between the

reference point and its nearest neighbors normals as:

$$(\mathrm{TBC_{NP}})^t = \frac{1}{|\mathcal{S}_a(\mathbf{r^t})|} \sum_{l \in \mathcal{S}_a(\mathbf{r^t})} \langle n(\mathbf{r}^t) \ , \ n(\mathbf{q}^l) \rangle, \tag{6.14}$$

where $\langle \cdot \ , \ \cdot \rangle$ is the dot product between the two vectors. Finally, the local predictions are thresholded and averaged along the border to get the estimation as the WSI level:

$$\mathrm{TBC_{NP}} = \frac{1}{N} \sum_{t=1}^{N} \mathbb{1}_{(\mathrm{TBC_{NP}})^t \geq \delta_{\mathrm{NP}}}. \tag{6.15}$$

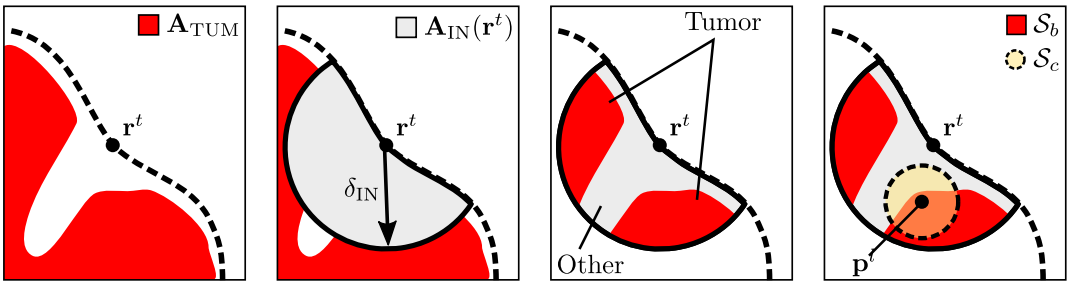
The hyperparameter $\delta_{\mathrm{NP}} \in [-1, 1]$ act as a confidence threshold. The larger δ_{NP} is, the more restrictive the metric is to consider the local estimation as a pushing border. For each point along the TB, we get a local estimation of either pushing ($\geq \delta_{\mathrm{NP}}$) or infiltrating ($< \delta_{\mathrm{NP}}$). Note that the local prediction are in range $(\mathrm{TBC_{NP}})^t \in [-1, 1]$. However, in practice, the dot products between the local components rarely sum up to a result lower than 0 as this would mean that the angle between the ideally pushing border and the raw estimate is, on average, greater than $\frac{\pi}{2}$ rad.

6.2.3 Tumor Ratio

For the second approach, we rely on local tumor tissue distribution. When moving along the TB, we can locally estimate the ratio of tumor tissue with respect to other tissues. If the ratio is high, we assume the local presence of tumor tissue is high, and then the TBC is more likely to be pushing. On the contrary, if we have a low presence of tumor tissue, we are more likely to have an infiltrating pattern. The quantity of tumor is assessed on the "inner" (*i.e.* toward the main tumor) side of the TB. The reason is that TB represents the delimitation of the tumor area and, therefore, properly surrounds it. A graphical overview for TBC based on local tumor ratio is given in Figure 6.7a-c. To know whether a point $\mathbf{p} \in \mathbf{A}$ is located on the so-called "inner" side of the margin, we use the border normals:

$$\Theta(\mathbf{p}) = \langle (\mathbf{p} - \mathbf{r}^t) \ , \ n(\mathbf{r}^t) \rangle \qquad \underset{t \in \{1,\dots,N\}}{\arg\min} \|\mathbf{p} - \mathbf{r}^t\|_2. \tag{6.16}$$

Here, if $\Theta(\mathbf{p}) < 0$, the point is located inside the tumor area. On the contrary, if $\Theta(\mathbf{p}) > 0$, the point lies outside of it. This estimation of elements' sidedness is only valid if the points of the TB are relatively close to each other. We can define the local inner



(a) Local point (b) Local mask (c) Local ratio (d) Local interaction

Figure 6.7 – Computation of TBC based on local tumor ratio and tissue interaction. (a) Point $\mathbf{r}^t$ on estimated pushing border and tumor area. (b) Mask $\mathbf{A}_{\text{IN}}(\mathbf{r}^t)$ with radius δ_{IN}. The mask is limited to the inner tumor area. (c) Masking of the tumor area. TBC is given as the ratio of tumor tissue within the mask. (d) TBC is given as the local interaction of tumor $\mathcal{S}_b$ with the surrounding tissues $\mathcal{S}_c$.

neighborhood $\mathbf{A}_{\text{IN}}(\mathbf{r}^t) = (a_{i,j}(\mathbf{r}^t))_{1\leq i\leq W\,,1\leq j\leq H}$ as:

$$a_{i,j}(\mathbf{r}^t) = \mathbb{1}_{(\Theta(\mathbf{p})\leq 0\,,\,\|\mathbf{p}-\mathbf{r}^t\|_2\leq\delta_{\text{IN}})} \qquad \mathbf{p} = \begin{pmatrix} i & j \end{pmatrix}^\top. \tag{6.17}$$

The value δ_{IN} acts as a threshold that selects the influence area of the metric. As we increase the value of δ_{IN}, we consider tissue areas that are further away from the TB. The TBC is then assessed for every point t along the border as the tumor ratio within the inner mask.

$$(\text{TBC}_{\text{RATIO}})^t = \frac{|\mathbf{A}_{\text{IN}}(\mathbf{r}^t) \cap \mathbf{A}_{\text{TUM}}|}{|\mathbf{A}_{\text{IN}}(\mathbf{r}^t)|}. \tag{6.18}$$

The local predictions are then thresholded and averaged along the TB to get the TBC at the WSI level:

$$\text{TBC}_{\text{RATIO}} = \frac{1}{N}\sum_{t=1}^{N} \mathbb{1}_{(\text{TBC}_{\text{RATIO}})^t\geq\delta_{\text{RATIO}}}, \tag{6.19}$$

where δ_{RATIO} is a threshold that fixes the ratio of tumor needed to consider the local area as pushing. The metric lies in the interval $[0, 1]$ where 0 and 1 mean infiltrating and pushing patterns, respectively.

6.2.4 Tumor Interaction

We introduce a last estimation of the TBC based on tumor interaction with neighbor tissues. One of the limitations of the previous approach is that we assume lower tumor density is correlated with an infiltrating pattern. However, a simple ratio cannot represent a tissue structure as it is solely based on total areas.

We propose to compute local tissue interaction [4]. A graphical representation is available in Figure 6.7a-b,d. Here, we start from the local masking $\mathbf{A}_{\text{IN}}(\mathbf{r}^t)$ as for the TBC based on tumor ratio. Given a tumor point that is part of the local mask, we can compute the number of its neighbors that are also part of the tumor area. As a result, if a tumor point is surrounded by tumor tissue, it is more likely to be part of a dense area (pushing). On the contrary, if a tumor point is isolated, it is more likely to be part of a sparse area (infiltrating). More formally, let $\{\mathbf{p}^l \in \mathbf{A}\}_{l=1}^{M}$ be the set of points that are part of the prediction map. We define the subset of points' index that belong to both the local mask and the tumor area:

$$\mathcal{S}_b(\mathbf{r}^t) = \{l \mid \mathbf{p}^l \in \mathbf{A}_{\text{IN}}(\mathbf{r}^t) \cap \mathbf{A}_{\text{TUM}}\} \qquad l \in \{1, \dots, M\}. \tag{6.20}$$

In addition, we define the local neighborhood of a point:

$$\mathcal{S}_c(\mathbf{p}^l) = \{k \mid \|\mathbf{p}^k - \mathbf{p}^l\|_2 \leq \delta_{\text{c}}\} \qquad k \in \{1, \dots, M\}, \tag{6.21}$$

where the value δ_{c} is a hyperparameter that limits the number of neighbors based on the distance to the source point. To assess the interaction between the tumor and the surrounding tissue, we go through the elements of $\mathcal{S}_c$ and check whether they also belong to the tumor area. As a result, for each point within the tumor mask area (*i.e.* $\mathbf{A}_{\text{IN}}(\mathbf{r}^t) \cap \mathbf{A}_{\text{TUM}}$) we get a local representation of the tumor compactness:

$$(\text{TBC}_{\text{INTER}})^t = \frac{1}{|\mathcal{S}_b(\mathbf{r}^t)|} \sum_{l \in \mathcal{S}_b(\mathbf{r}^t)} \frac{1}{|\mathcal{S}_c(\mathbf{p}^l)|} \sum_{k \in \mathcal{S}_c(\mathbf{p}^l)} \mathbb{1}_{(\mathbf{p}^k \in \mathbf{A}_{\text{TUM}})}. \tag{6.22}$$

The value is averaged along the border to get the TBC estimation at the WSI level:

$$\text{TBC}_{\text{INTER}} = \frac{1}{N} \sum_{t=1}^{N} \mathbb{1}_{(\text{TBC}_{\text{INTER}})^t \geq \delta_{\text{INTER}}}, \tag{6.23}$$

where δ_{INTER} is the metric specific threshold. The local metric values lie in the interval $[0, 1]$ where 0 and 1 mean infiltrating and pushing patterns, respectively. A representation

of the introduced metrics' behavior on toy examples is available in the supplementary section E.5.

6.3 Extra Definitions

In this section, we tackle less well-established metrics. We first introduce the definition of TMR in subsection 6.3.1 and then move to tissue representation around the tumor border in subsection 6.3.2.

6.3.1 Tumor to Mucin Ratio

TMR is indirectly reported in routine diagnostic to identify mucinous adenocarcinoma in CRC. If the mucinous area represents more than 50% of the overall tumor area, the case is considered mucinous. Recent work uses a machine learning-based approach to quantify the presence of mucin in CRC [107]. Unfortunately, the method still needs manual input from the pathologist to locate the final area.

We propose using our segmentation method to compute TMR fully automatedly. Given a WSI, we assume a segmentation map is generated where classes TUM and mucin (MUC) are available. The tumor $\mathbf{A}_{\text{TUM}}$ and mucin $\mathbf{A}_{\text{MUC}}$ areas are defined as the areas that are segmented by the model as tumor and mucin, respectively. Given the introduced components, we define TMR at the WSI level as:

$$\text{TMR} = \frac{|\mathbf{A}_{\text{TUM}}|}{|\mathbf{A}_{\text{TUM}} \cup \mathbf{A}_{\text{MUC}}|}. \tag{6.24}$$

Note that $\mathbf{A}_{\text{MUC}}$ also includes mucin from normal tissue crypts. However, we assume the amount is negligible at the WSI level.

6.3.2 Stroma Tissue Distribution

Another interesting feature is the distribution of tissue classes along the tumor border. It is defined as the presence of a specific tissue class in the TB surroundings. For example, when looking at CRC cases where the tumor reached the muscle layer (*i.e.* pT3), we can investigate tissue distribution at the interface. Sometimes, we observe close contact between the two tissues where the tumor directly penetrates the muscle area. In other cases, we see a broad band of TA-STR that acts as a boundary between them. As TA-STR has been proven to be an interesting prognostic factor [109], we can wonder how the presence or absence of TA-STR at the interface influences survival predictions.

The assessment of tissue distribution cannot be visually done. Fortunately, based on our

estimation of the TB, we can efficiently compute tissue distribution across large cohorts. To do so, we define the outer region $\mathbf{A}_{\mathrm{OUT}}(\mathbf{r}^t) = (a_{i,j}(\mathbf{r}^t))_{1\leq i\leq W\,,1\leq j\leq H}$ centered in $\mathbf{r}^t$ of the TB as:

$$a_{i,j}(\mathbf{r}^t) = \mathbb{1}_{(\Theta(\mathbf{p})>0\,,\,\|\mathbf{p}-\mathbf{r}^t\|_2\leq\delta_{\mathrm{OUT}})} \qquad \mathbf{p} = \begin{pmatrix} i & j \end{pmatrix}^\top, \tag{6.25}$$

The mask represents the region of the tumor that is progressing toward normal tissue. For each location, we check for the presence of STR within the range of the TB. As the distance to the border is small, we can assume the TA-STR and STR classes overlap in terms of definition. A threshold is then applied to the local metrics, and the results are averaged along the TB as:

$$\mathrm{TD}_{\mathrm{STR}} = \frac{1}{N}\sum_{t=1}^{N} \mathbb{1}_{(\mathrm{TD}_{\mathrm{STR}})^t\geq\delta_{\mathrm{TD}}} \qquad \text{and} \qquad (\mathrm{TD}_{\mathrm{STR}})^t = \frac{|\mathbf{A}_{\mathrm{OUT}}(\mathbf{r}^t)\cap\mathbf{A}_{\mathrm{STR}}|}{|\mathbf{A}_{\mathrm{OUT}}(\mathbf{r}^t)|}. \tag{6.26}$$

The metric gives an overall value of the presence of stroma at the border. A value of 1 indicates a systematic presence of stroma along the border. On the contrary, 0 means no stroma is present and that the tumor directly grows through the normal tissues. This work focuses on the TA-STR component. However, the metric can be applied to different tissue classes such as adipose (*e.g.* check for the presence of fat tissue), lymphocytes (*e.g.* check for immune response/inflammation), or debris (*e.g.* presence of necrotic tissue).

6.4 Experiments

In this section, we evaluate the performance of the metrics on multiple cohorts. First, we define the experimental setup in subsection 6.4.1. We then validate our result on TSR along with manual annotations in subsection 6.4.2. Next, we compare the three automated TBC prediction approaches to expert annotations in subsection 6.4.3. Afterward, we analyze the correlation of TMR and tissue distribution with other clinical parameters. Finally, we validate the metrics estimation by performing univariate and multivariate survival analyses on multiple cohorts in subsection 6.4.5-6.4.6.

6.4.1 Experimental Settings

For the experiment, we use the coarse to refined (C2R) detection model presented in chapter 5 to generate fine-grained tissue segmentation maps. The selected architecture implements multi-level feature extraction, visual consistency through l^1 regularization, and self-correlation map prediction.

The interaction distance is set to $\delta_{\text{TA-STR}} = 1000\mu\text{m}$ thus including more context for metric estimations [106]. We use a 50% cutoff to split patients into two groups based on the TSR measure as *high* (TSR $\geq 50\%$) and *low* (TSR $< 50\%$).

The identification of TSR ROI is performed using $L = 6$ sub-filters to ensure the presence of surrounding tumor tissue. The size of the filter is fixed based on the size of the $10\times$ lens, which corresponds to a circle of 2500µm in diameter. Based on the resolution of the segmentation algorithm (*i.e.* $\simeq 15\mu\text{m/px}$), we end up with a size of $N = 161$ for the filters. Moreover, to design the detection filter Θ, we use parameters $a = \frac{\pi}{8}$, $b = 0.1N$, $c = \{0, \frac{\pi}{9}, \frac{2\pi}{9}\}$. We use multiple offset values c to improve the detection of ROIs and cover all possible orientations. We compute the filtering with each value of c and aggregate the results as the union of all detected regions. The value for the tumor threshold in the ROI is empirically fixed to $\delta_{\text{ROI}} = 0.25$. For the average filter Φ, we use the same filter size N as for the ROI candidates. We use $K = 3$ to compute the ROI estimations at the WSI level.

The detection of the TB is done by fixing the value of the decision function to $\Delta_{\text{DB}} = -0.2$. It ensures that the decision line lies closer to the main tumor area. The selection of metric thresholds for TBC (*i.e.* δ_{NP}, δ_{RATIO}, and δ_{INTER}) are discussed in the next section. For the sake of simplicity, the distances of the inner and outer regions are fixed to the same value as the distance to TA-STR as $\delta_{\text{TA-STR}} = \delta_{\text{IN}} = \delta_{\text{OUT}} = \delta_{\text{TD}} = 1000\mu\text{m}$. Regarding local interaction, we empirically set the local neighborhood to $\delta_{\text{c}} = 150\mu\text{m}$. The idea is to keep $\delta_{\text{c}} \ll \delta_{\text{TA-STR}}$ to get a micro versus macro representation of the area.

Note that the convolution of large filters over WSI segmentation maps can be a computationally expansive task. If the segmented area is too large, we recommend downsampling the predictions map to make a first coarse estimation of the ROIs. The detection can then be refined by applying filtering at full resolution on coarsely detected areas. Another solution is to perform chunk-wise analysis to reduce computational complexity.

The result are validated on five CRC patients cohorts ($\mathcal{P}_A$, $\mathcal{P}_B$, $\mathcal{P}_C$, $\mathcal{P}_D$, and $\mathcal{P}_E$). We have access to overall and/or disease-free survival for each cohort patient. Patients who underwent preoperative treatment are excluded from the data. Please refer to subsection 2.6.2 for extended information about available patient data.

6.4.2 Automated TSR Evaluation

To validate the performance of our TSR estimation, we rely on 10 annotated WSIs from the University of Southern Denmark (SDU). For each slide, we have access to the ROI location selected by the expert as well as its label: *high* or *low*. We compute the TSR metrics (*i.e.* TSR_{WSI}, TSR_{TA}, TSR_{ROI1}, TSR_{ROI2}, TSR_{ROI3}, and $\text{TSR}_{\overline{\text{ROI}}}$) on all slides. The value TSR_{ANNO} is given as the TSR evaluated given the ROI selected by the expert. The distributions of the metrics are presented in Figure 6.8. We display the Pearson

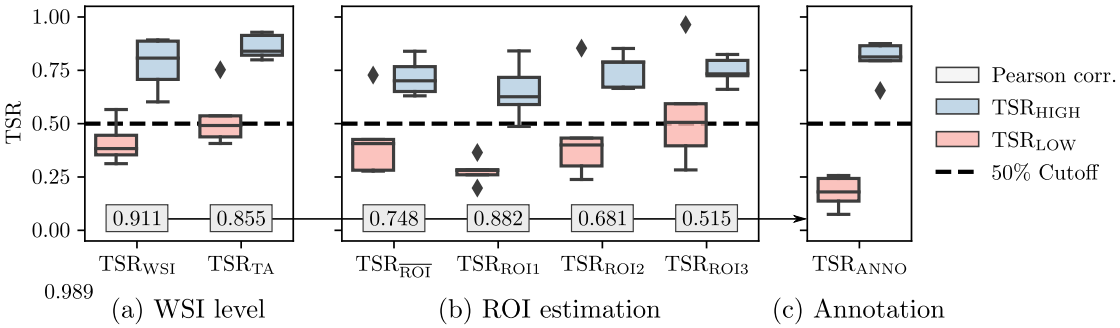


Figure 6.8 – Comparison of TSR prediction based on different approaches. We use a 50% cutoff to split between TSR high (blue) and low (red). We report the Pearson correlation (gray) between automated predictions and the annotated area. (a) Estimation at the WSI level with and without TA-STR. (b) Detection of the top $K = 3$ ROI and averaged results. (c) TSR estimated on manually annotated ROIs.

correlation between the annotated area and the automated evaluations.

When looking at the distribution of the annotated area, we observe a distinct split between the high and low groups. The fact that the estimation of TSR within the selected area matches the ground truth group (*i.e.* perfect split at 50%) is an indicator that the segmentation model is able to properly assess TUM and STR areas.

For the WSI level metrics, we can see that the distribution is shifted toward high TSR. This is explained by the fact that the diagnostic slides tend to include large tumor areas. When limiting the computation of TSR to TA-STR, we get a better split between the two groups. By doing so, we reduce the impact of distant normal stroma. As a result, we get a better split between the two groups. However, the influence of the tumor area is increased, and the overall TSR estimation is even more biased toward high predictions.

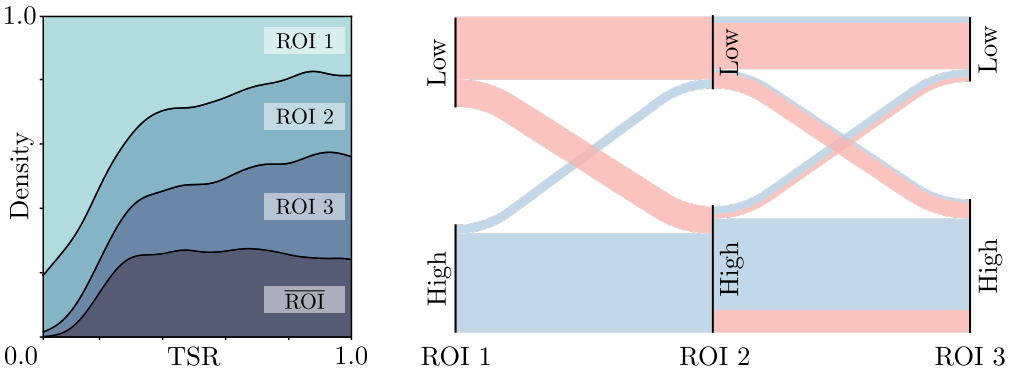
Regarding the estimation of the ROI, we observe a slowly shifting mean as we increase the number of top K ROIs. It is explained by the fact that the detection of ROIs is based on the presence of dense stroma regions. The areas with the highest stroma concentration are selected first (*i.e.* first ROI). As a result, the remaining regions have, by definition, less stroma and hence higher TSR. The averaging of top K areas produces TSR estimations that are roughly centered around the decision boundary (*i.e.* 50%). However, due to the small size of the validation cohort, we must be careful about generalizing the results to larger cohorts. A visualization of the generated output ROIs on a WSI is available in the supplementary material in section E.2.

The detection of TSR is applied to all cohorts $\mathcal{P}_{A-E}$. When processing new WSIs, the detection of TSR may face challenges. For example, the approach can fail to identify a ROI that meets the "four directions" for tumor detection. Moreover, it is not always possible to detect K ROIs depending on the size of the WSI and tumor area. As a result,

Table 6.1 – Error rate of the automated TSR detection approaches for each cohort. We display the total number of slides with successful detection and the error rate as a percentage.

Metric (ER %)	$\mathcal{P}_A$	$\mathcal{P}_B$	$\mathcal{P}_C$	$\mathcal{P}_D$	$\mathcal{P}_E$	$\mathcal{P}_{A-E}$
Slides	739 (0.0)	174 (0.0)	556 (0.0)	469 (0.0)	118 (0.0)	2055 (0.0)
Slide-level						
TSR_{WSI}	739 (0.0)	174 (0.0)	556 (0.0)	469 (0.0)	117 (0.8)	2055 (0.0)
TSR_{TA}	739 (0.0)	174 (0.0)	506 (0.0)	440 (0.0)	117 (0.8)	2055 (0.0)
ROI level						
TSR_{ROI1}	725 (1.9)	174 (0.0)	556 (9.0)	469 (6.2)	116 (1.7)	1961 (4.6)
TSR_{ROI2}	717 (3.0)	171 (1.7)	487 (12.4)	409 (12.8)	116 (1.7)	1900 (7.6)
TSR_{ROI3}	704 (4.7)	167 (4.0)	460 (17.3)	373 (20.5)	115 (2.5)	1819 (11.5)
$\text{TSR}_{\overline{\text{ROI}}}$	725 (1.9)	174 (0.0)	556 (9.0)	469 (6.2)	116 (1.7)	1961 (4.6)

Abbreviations: Error rate as a percentage (ER %).

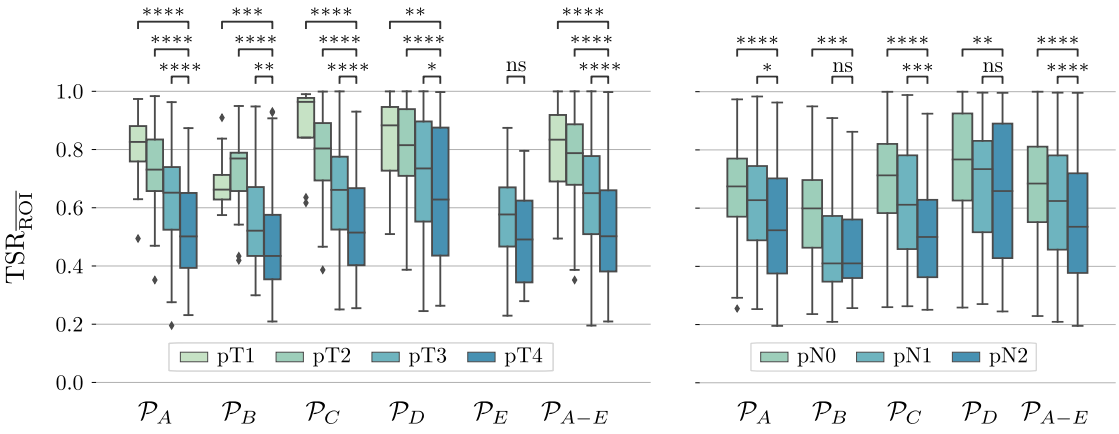


(a) TSR densities (b) TSR transition

Figure 6.9 – Distribution and evolution of TSR estimation for cohort $\mathcal{P}_{A-E}$. (a) Distributions of TSR values based on ROI metrics. (b) Transition of TSR estimation between low and high group as we include additional ROIs.

an essential feature of our tool is the error rate (ER). It quantifies how likely our model is to fail TSR estimation. In Table 6.1, we display the ER of the different TSR detection approaches on WSIs. We get a perfect detection rate when computing TSR detection at the WSI level. Regarding the ROI-based methods, the ER tends to increase with the number of regions. For $\mathcal{P}_A$, $\mathcal{P}_B$, and $\mathcal{P}_E$ we manage to extract $K = 3$ ROIs in a large majority of cases. However for $\mathcal{P}_C$, and $\mathcal{P}_D$ the ER goes up to 20.3% for the third ROI. It mainly comes from the fact that the WSI images are small, prohibiting tumor areas from being extracted using a 2.5mm lens. Part of the detection errors also come from high-grade examples where the model tends to classify tumor tissue as STR. These data limitations must be considered when performing TSR estimation.

We further explore the behavior of TSR by looking at the distribution and transition of the metrics on larger cohorts in Figure 6.9. We first display the reparation of TSR



(a) Tumor depth (b) Positive lymph nodes

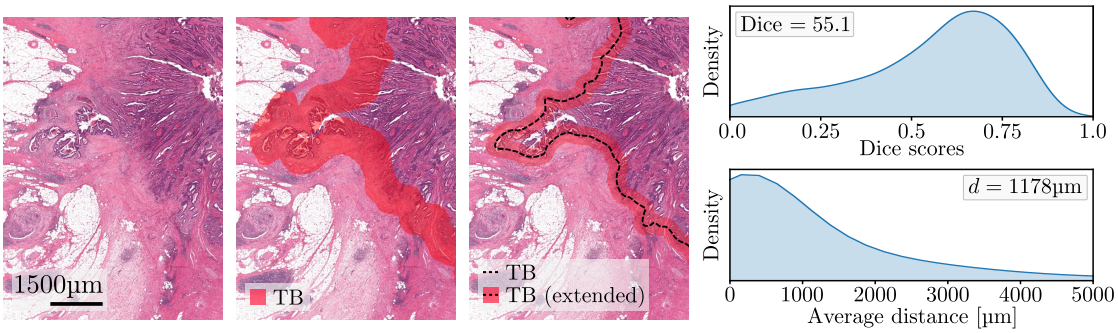
ns: $p \geq 0.05$, *: $p \leq 0.05$, **: $p \leq 0.01$, ***: $p \leq 0.001$, and ****: $p \leq 0.0001$

Figure 6.10 – Correlation of $\text{TSR}_{\overline{\text{ROI}}}$ with clinical feature other multiple cohorts ($\mathcal{P}_A$ to $\mathcal{P}_E$). We report the correlation with respect to the (a) depth of invasion and (b) number of positive lymph nodes. Statistical significance is tested using the Mann–Whitney U test.

across ROIs. We observe that the first detected ROI tend to have lower TSR values. As we include more ROI, we increase the average detected TSR. It confirms our previous results with manually labeled areas. The second plot shows the transition of the TSR value when applying a 50% cutoff. For each patient, we compute the change in TSR group (low and high) as we include more ROIs. For example, a patient without state transition (low-to-high or high-to-low) has a constant line. On the contrary, a patient with group variations will show state transitions. The results highlight the unbalanced aspect of state transitions. We observe few changes from high-to-low TSR compared to low-to-high. Roughly half of the patients whose first region is labeled as TSR low have their third ROI labeled as high. We assume that taking the average between the three regions might dampen this effect.

In Figure 6.10, we present the correlation between ROI-averaged TSR estimates (*i.e.* $\text{TSR}_{\overline{\text{ROI}}}$) and clinical feature. The correlation is assessed on all cohorts, and statistical significance is tested using the Mann–Whitney U test as we expect non-Gaussian distributions. Out of all the clinical features available, we display the depth of invasion (pT) and the positive lymph nodes (pN). For additional results on other TSR estimates and clinical features, please refer to supplementary material in section E.4.

We observe a strong correlation between the TSR estimation and the clinical features. The depth of invasion is inversely proportional to the TSR estimation. As the tumor progresses through the tissue, it tends to have a higher TA-STR density. More interestingly, the positive lymph node assessment is correlated with the TSR. A high pN value indicates lower TSR estimation. The results are expected as both pT and pN are used to grade



(a) WSI crop (b) Annotation (c) Prediction (d) Distribution on $\mathcal{P}_C$

Figure 6.11 – Comparison of manual and automated TB. (a) Close view of the WSI tumor border. (b) Expert annotation. (c) TB estimation (red: dilated). (d) Distribution of Dice scores and average distance to annotation on $\mathcal{P}_C$. We report the median across slides.

tumor stage, and high cancer stages are linked to more invasive and aggressive patterns.

6.4.3 Automated TBC Evaluation

Before validating the performance of the TBC, we need to assess the quality of the TB estimation. We take advantage of 532 manually labeled TB by expert pathologists on $\mathcal{P}_C$ cohort. The borders are drawn using a digital brush marker. An example of an annotated area with estimated TB is displayed in Figure 6.11. We select two measures to quantify the closeness of our prediction to the annotations. The first is the spatial consistency (DSC) score between the predicted line and the labeled area. The second metric measures the average minimal distance between the two predictions. During the evaluation process, the predicted TB is dilated to reach the thickness of the annotation to allow a fair estimation.

Overall, we get 55.1% and 1,178µm for DSC and average distance, respectively. When looking at the distributions of DSC scores, we can observe that most predictions achieve a high score. After manually reviewing the cases, we observe four main causes of erroneous TB detection by our model. The first reason is the annotations' lack of consistency in including or excluding the TB interface between normal mucosa and tumor. The second reason is the presence of large muscle or adipose blobs within the main tumor area. It is difficult for our approach to distinguish between a healthy tissue encapsulated within the main tumor and a proper TB linked to tumor progression. Such cases are hard to visually assess as the delimitation of the TB depends on the 3D tissue structure, which is lost after cutting. Thirdly, our model fails to capture small tumor buds in highly inflamed areas. As a result, we tend to prioritize regions with dense tumor blobs when estimating TB. Finally, we observed that distant mucin blobs (*i.e.* disconnected from the main tumor) are not included in the main tumor area by our model. The reason is

Table 6.2 – Number TBC annotations available across cohorts. We rely on two different groups of annotations: TBC-Patient (assessed on multiple slides and averaged), TBC-Slide (assessed on a single slide).

Characteristics	$\mathcal{P}_A$ ($n = 383$)	$\mathcal{P}_B$ ($n = 174$)	$\mathcal{P}_D$ ($n = 463$)	$\mathcal{P}_{A,B,D}$ ($n = 1020$)
TBC-Patient(%)				
Pushing	149 (54.6%)	89 (53.0%)	-	238 (54.0%)
Infiltrating	124 (45.4%)	79 (47.0%)	-	203 (46.0%)
TBC-Slide(%)				
Pushing	-	62 (37.1%)	96 (35.7%)	158 (36.2%)
Infiltrating	-	105 (62.9%)	173 (64.3%)	278 (63.8%)

Table 6.3 – Detection performance of the automated TBC approaches for each cohort. We display the total number of slides with successful detection, the error rate as a percentage, and the detected tissue layers.

Metric (ER %)	$\mathcal{P}_A$	$\mathcal{P}_B$	$\mathcal{P}_C$	$\mathcal{P}_D$	$\mathcal{P}_E$	$\mathcal{P}_{A-E}$
Slides	739 (0.0)	174 (0.0)	556 (0.0)	469 (0.0)	118 (0.0)	2055 (0.0)
Tissue						
Adipose	710 (3.9)	163 (6.3)	538 (3.2)	208 (55.7)	114 (3.4)	1733 (15.7)
Muscle	713 (3.5)	172 (1.1)	556 (0.0)	252 (46.3)	112 (5.1)	1804 (12.2)
Adi. | Mus.	736 (0.4)	174 (0.0)	556 (0.0)	267 (43.1)	117 (0.8)	1849 (10.0)
Tumor	738 (0.1)	174 (0.0)	547 (1.6)	467 (0.4)	117 (0.8)	2043 (0.6)
TBC (any)	736 (0.4)	174 (0.0)	542 (2.5)	248 (47.1)	117 (0.8)	1817 (11.6)

Abbreviations: Error rate as a percentage (ER %).

that our approach uses a region-growing pattern to estimate the main tumor blob.

To assess the quality of the TBC estimation, we rely on annotations by expert pathologists. The TBC is scored using 10% increments from infiltrating (0%) to pushing (100%). When used for survival prediction or correlation, the TBC is usually stratified based on a 50% threshold as pushing ($\geq$ 50%) and infiltrating ($<$ 50%). The number of available annotations is reported in Table 6.2. We distinguish two groups of annotation as TBC-Patient and TBC-Slide. In the first case, the TBC is given at the patient level. It is computed on multiple slides and then averaged. In this setting, we cannot access the slides used to evaluate the metric. Moreover, the annotations are generated by multiple experts. In the second case, all cases are reviewed by the same pathologist. The TBC is assessed on a single diagnostic slide. Here, we have a one-to-one slide comparison between annotated and automated TBC.

The automatic detection of TBC is applied to cohorts $\mathcal{P}_{A-E}$. We compute the ER for the TBC estimation as well as for the identification of the tissue layers used to compute TBC (*i.e.* muscle, adipose, and tumor). The results are shown in Table 6.3. We observe that in a large majority of cases, the model is able to identify either muscle or adipose tissue and thus estimate the tumor growing direction (ER = 10.0%). The exception is

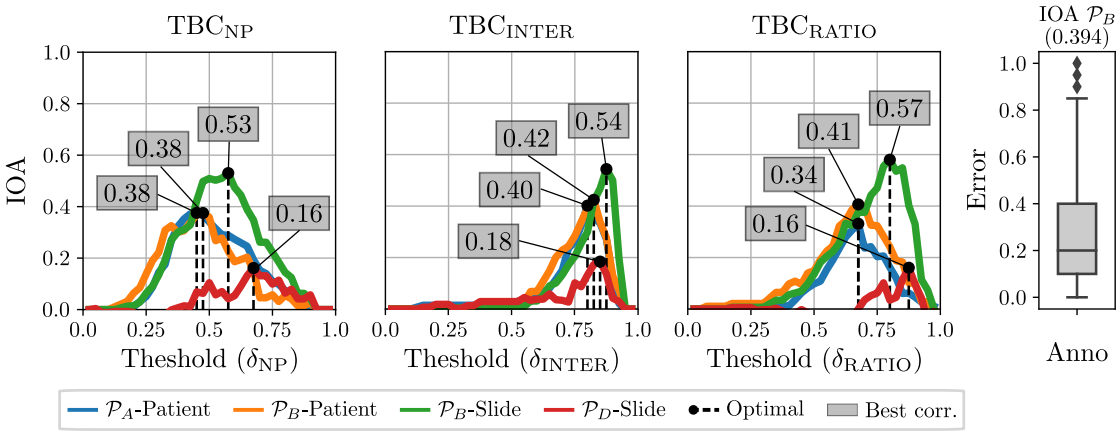
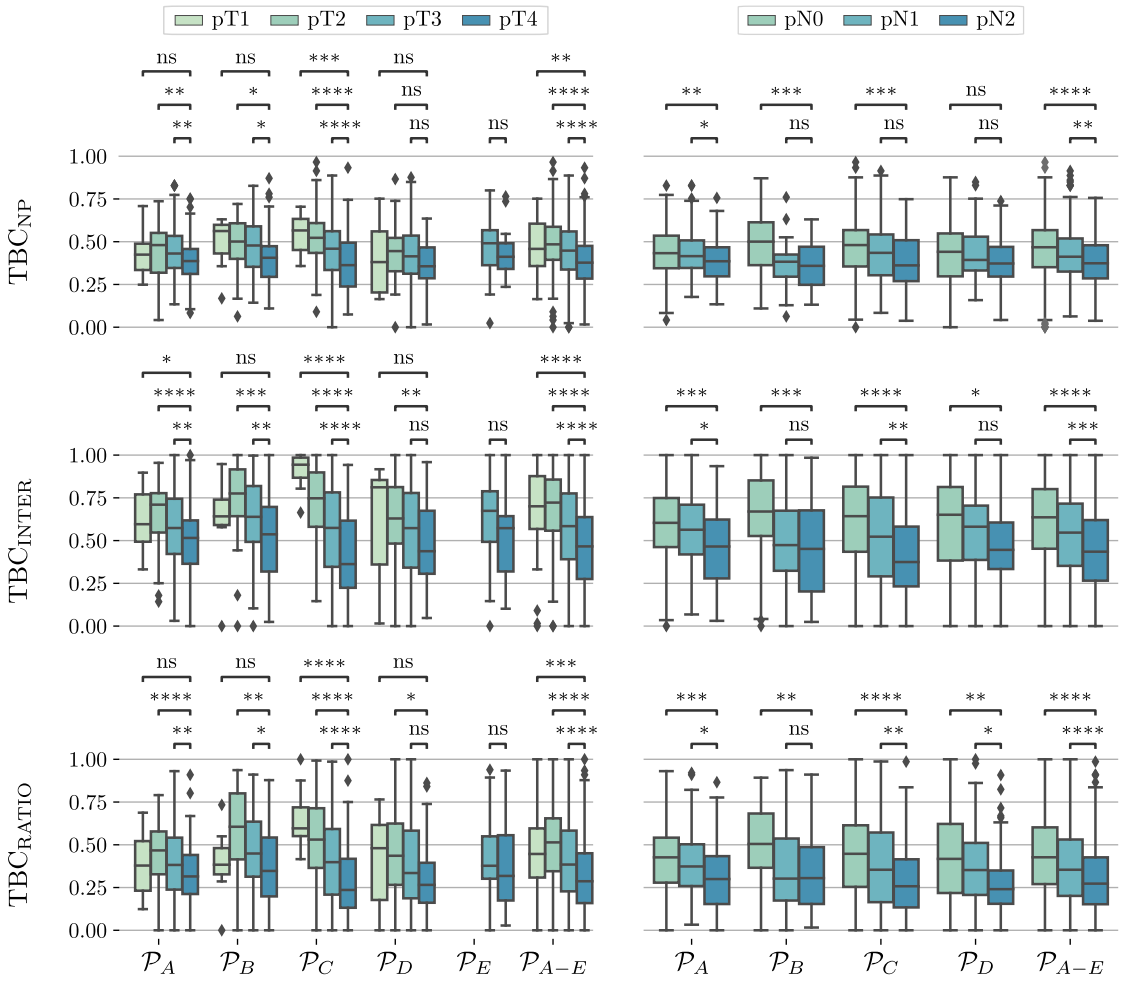


Figure 6.12 – Evolution of interobserver agreement (IOA) between TBC estimations and expert's annotations as a function of the metric threshold. We display each cohort's optimal threshold and IOA. The last plot shows the absolute difference on TBC estimation between annotators $\mathcal{P}_B$-Patient and $\mathcal{P}_B$-Slide, as well as the resulting IOA on top of the figure.

$\mathcal{P}_D$, where we identify a significant drop in performance. In almost half of the WSIs, the model fails to detect muscle or adipose tissue and, therefore, the orientation of the tumor. This is mainly because WSIs only include the tumor area and hence lack contextual information. Regarding the detection of TBC, we achieve a ER of 11.6% which goes down to 1.1% if $\mathcal{P}_D$ is omitted. Note that in some cases, the detected tumor border is not large enough to compute relevant statistics and is, therefore, discarded.

In Figure 6.12, we assess the performance of the automated TBC estimation with respect to the expert annotations. We report Cohen's kappa score, also known as IOA. The metric measures the agreement between two annotators, where 0 and 1 mean no and full agreement, respectively. For the annotation, we use 50% as a split between TBC high and low. For the automated approaches, we swipe the thresholds (*i.e.* δ_{NP}, δ_{INTER}, and δ_{RATIO}). In the far right plot, we display the IOA within the $\mathcal{P}_B$ cohort where we can compare TBC annotation between TBC-Patient and TBC-Slide.

We observe a large variance in terms of IOA evolution across the different metrics. Each metric shows a different operating range. We get the highest results in the intervals $[0.45, 0.65]$, $[0.8, 0.925]$, $[0.675, 0.875]$ for δ_{NP}, δ_{INTER}, and δ_{RATIO}, respectively. For the δ_{NP} and δ_{RATIO}, we perceive a Gaussian-like distribution of the IOA for all cohorts. This observation is not valid for the distributions of the δ_{INTER} metric where the distributions appear skewed. We have to be careful while selecting the final operating threshold. Choosing a threshold that is too high might result in a bad generalization of the metrics to other cohorts, as it would be too selective. Here, a reasonable approach would be to consider thresholds that maximize the average correlation as $\delta_{NP} \simeq 0.45$, $\delta_{INTER} \simeq 0.8$, $\delta_{RATIO} \simeq 0.675$. The best IOA is achieve on $\mathcal{P}_B$-Slide using TBC$_{RATIO}$ with a score of



(a) Tumor depth

(b) Positive lymph nodes

ns: $p \geq 0.05$, *: $p \leq 0.05$, **: $p \leq 0.01$, ***: $p \leq 0.001$, and ****: $p \leq 0.0001$

Figure 6.13 – Correlation of automated TBC estimation (δ_{NP}, δ_{INTER}, δ_{RATIO}) with clinical feature on multiple cohorts ($\mathcal{P}_A$ to $\mathcal{P}_E$). We report correlation with (a) depth of invasion and (b) number of positive lymph nodes. Statistical significance is tested using the Mann–Whitney U test.

0.57.

When looking at the performance of the metrics cohorts-wise, we can also observe large variations. The $\mathcal{P}_B$-Slide results have the best outcomes across all metrics. Here, we have access to the original slides used for annotation and can perform one-to-one evaluation. The fact that we have high IOAs means the model can properly assess TBC in an automated fashion. The evaluation on $\mathcal{P}_{A,B}$-Patient shows similar but lower results. The consistency across the cohorts is expected as they both are from the same institute and are annotated by the same experts. Moreover, as we do not have access to the same patients'

slides as the expert annotator, we cannot ensure a one-to-one correspondence between the features, which can explain the performance drop. Still, we achieve reasonable IOA values further validates our approaches. More surprisingly, the agreement on $\mathcal{P}_D$ is relatively low compared to other cohorts. This can be explained by the fact that the samples in the cohort have lower quality and, thus, are more challenging to assess for both the expert and the model. Moreover, we observed inconsistencies in the annotations that can further explain the low agreement. Finally, we compare the IOA between the two groups on the $\mathcal{P}_B$ cohort. The outcome achieves an IOA score of 0.394, which is considered as low. It shows that precise TBC estimation is challenging when relying on a single patient slide.

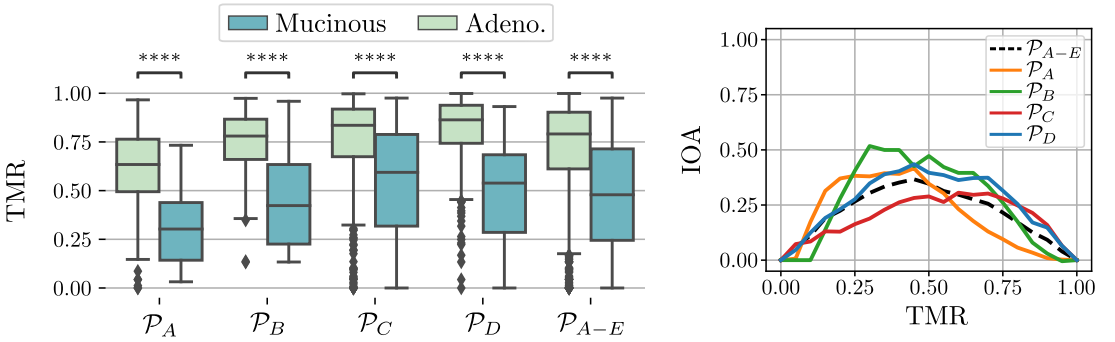
Finally, we investigate the performance of our automated approaches by checking the correlation of predictions with clinical features in Figure 6.13. TBC_{NP} shows lower correlation with pT and pN compared to $\text{TBC}_{\text{INTER}}$ and $\text{TBC}_{\text{RATIO}}$. We see the same overall behavior for all metrics where higher pT and pN correlate with lower TBC (infiltrating pattern). The metrics tend to fail to show statistical relevance between pT1 and pT4. This can be explained by the fact that pT1 is underrepresented with only a few cases (*i.e.* 3% of all patients). Another observation is the absence of statistical significance between pN1 and pN2 when correlated to TBC. As the difference between the two metrics is based on the number of positive lymph nodes (from one to three for pN1 and more than three for pN2), their patterns might share similarities.

We compare the performance of the metrics applied to our previous Self-Rule to Multi Adapt (SRMA) model. The evaluation of the correlation between the use of SRMA and C2R models on TSR estimation, as well as the correlation of automated prediction with pathologists annotation for SRMA, are available in section E.3.

6.4.4 Extra Metrics

We investigate the additional metrics. In diagnosis, if more than 50% of the tumor volume is occupied by mucin, the case is considered mucinous [93]. As TMR is inversely proportional to the amount of mucin, adenocarcinoma and mucinous cases should be connected to high and low TMR, respectively. The results are presented in Figure 6.14.

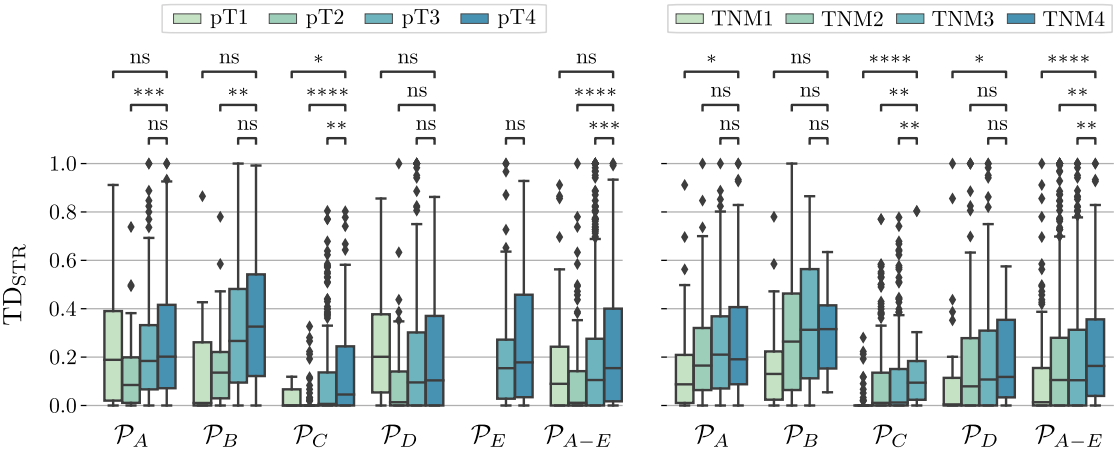
The first plot highlights th correlation between the two components. We observe a clear difference between the groups for all presented cohorts. In the second plot, we apply different thresholds to the automated TMR prediction and compute IOA with histological types. We observe that for all datasets, the optimal threshold is located close to the decision value (*i.e.* 50%). However, the overall IOA still remains low. After manually reviewing the results, it appears that the lack of patient slides causes a drop in performance. In most cases, the model can properly assess TMR at the WSI level. However, more than one slide is needed to fix the patient histological type.



(a) Correlation with histological type (b) IOA with histological type

ns: $p \geq 0.05$, *: $p \leq 0.05$, **: $p \leq 0.01$, ***: $p \leq 0.001$, and ****: $p \leq 0.0001$

Figure 6.14 – Correlation between TMR automated prediction and histological types. (a) Distribution across the different datasets with CRC type split. (b) IOA between TMR and labeled histological type for different TMR thresholds.



(a) Tumor depth (b) Cancer stage

ns: $p \geq 0.05$, *: $p \leq 0.05$, **: $p \leq 0.01$, ***: $p \leq 0.001$, and ****: $p \leq 0.0001$

Figure 6.15 – Correlation of stroma tissue distribution TD_{STR} with (a) depth of invasion and (b) cancer classification across five different cohorts. The p-value thresholds are displayed at the bottom of the table.

Regarding tissue distribution, we follow the same logic and check for correlation with other clinical features. For the selection of the threshold, no recommendation exists. As a result, the threshold is selected such that we equally split patients into low and high groups (*i.e.* median of the metric) as 20% ($\delta_{TD} = 0.2$). The results for stroma local distribution at the tumor boundary are presented in Figure 6.15. For both clinical metrics, we observe a correlation with TD_{STR}. As expected, high cancer stages and deeper invasions are more prone to have high stroma tissue at the boundary.

6.4.5 Univariate

With the previous results, we are now reasonably certain that the generated outputs align with the expert annotations. The next phase is to take advantage of the automated pipeline to predict survival. To do so, we use five different cohorts where we have access to both overall survival (OS) and disease-free survival (DFS) data. We remove patients who died up to three months after surgery. Regarding the study length, we have access to patient follow-up up to 10 years after surgery. However, the median follow-up time across all cohorts is around five years. As a result, we use the gold standard 5-year period for the event analysis. We estimate hazard ratios using Cox proportional hazards (CPH) with l^1-norm regularization and penalty factor $\lambda = 0.01$. The results are presented in Figure 6.16 for OS and DFS. We report the main clinical metrics as gender, depth of invasion (pT), lymph node metastases (pN), cancer classification (TNM), histological type (adenocarcinoma or mucinous), microsatellite instability (MSI), tumor budding, lymphatic invasion, venous invasion, and manual TBC estimation. Along with the clinical labels, we report the output of the automated approaches for TSR (*i.e.* TSR_{WSI}, TSR_{TA}, $\text{TSR}_{\overline{\text{ROI}}}$, and TSR_{ROI1}), TBC (*i.e.* TBC_{NP}, $\text{TBC}_{\text{RATIO}}$, and $\text{TBC}_{\text{INTER}}$), and extra (*i.e.* TMR, and TD_{STR}). For the automated approaches, we use a threshold at 50% to split continuous variables into low and high groups. The exception is TSR_{TA}, where we use the group median across all cohorts instead. The reason lies in the definition of the tissue distribution that is not correlated with an existing clinical variable and hence is not centered around the 50% decision boundary.

When focusing on the clinical variable, we observe that lymph node metastasis, lymphatic invasion, and cancer staging show the best predictive value for both OS and DFS. For tumor depth of invasion, we get mixed results. The reason is that we rely on a few pT1-2 examples. Regarding gender, we get a statistically significant hazard ratio where women tend to have a lower hazard probability. Here, we must be careful with interpreting results as the data are not adjusted for age [98, 149]. We also note that budding is correlated to DFS. Moreover, the manual annotation for TBC achieves statistical significance on the first set for OS and DFS.

Regarding the automated approaches, TSR estimation shows similar results across all cohorts. The evaluations of the TSR based on ROIs (*i.e.* $\text{TSR}_{\overline{\text{ROI}}}$, and TSR_{ROI1}) achieve relevant result for both OS and DFS. When it comes to TSR assessment at the WSI level, the results are mixed. While we can observe a difference in the hazard ratio for DFS, we get no clear outcome for OS. In addition, we do not see a clear difference between $\text{TSR}_{\overline{\text{WSI}}}$, and TSR_{TA}. Hence, it is not possible to validate our previous intuition that considering TA-STR for TSR estimation would help the model for patient stratification.

For TBC estimation, all three approaches manage to reach statistical significance for both OS and DFS when merging the cohorts. We observe that the $\text{TBC}_{\text{INTER}}$ predictor shows a more coherent behavior across cohorts. The manual annotation highlights a

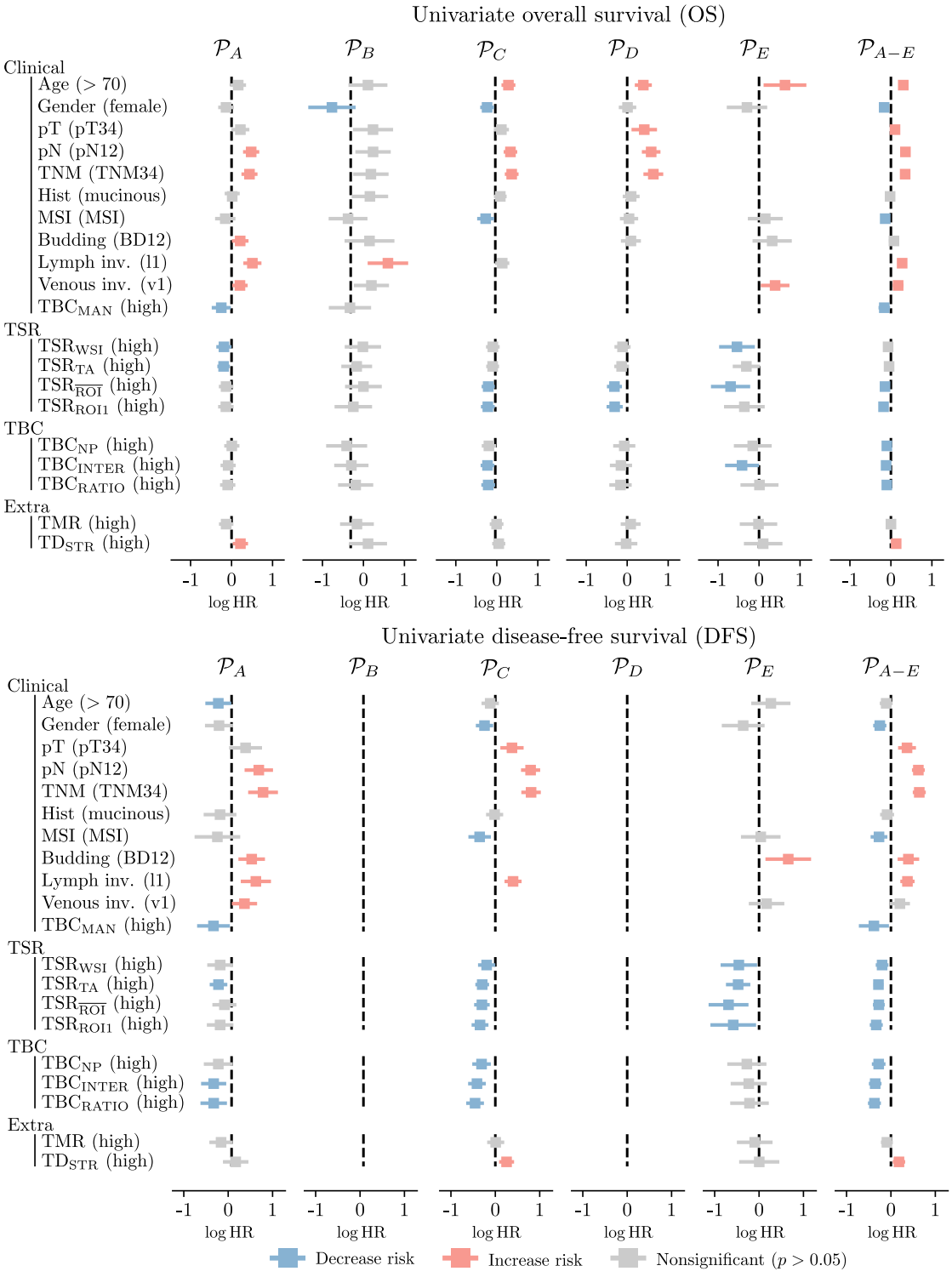


Figure 6.16 – Univariate CPH estimation for OS and DFS based on clinical and automated metrics across cohorts. We consider a 5-year period for the study where features are binarized before model fitting. The selected group is indicated between parentheses.

difference in the hazard ratio for OS in $\mathcal{P}_A$, which is not the case for the automated approaches. This is most likely because the manual TBC is assessed on multiple slides, whereas the automated predictor uses a single WSI to assess TBC. More specifically, when looking at the TBC estimation on the stage II cohort (*i.e.* $\mathcal{P}_E$) we do not perceive any divergences between the groups. It suggests that TBC does not help stage II CRC patient stratification.

Finally, we focus on the additional metrics. The TMR does not show any difference in predictions between the two groups. This outcome confirms the previous results obtained with the histological type, where we could not see any variations between adenocarcinoma and mucinous types. Concerning the distribution of stroma at the boundary (TD_{STR}), only one of the cohorts shows a distinction between high and low categories.

We provide the Kaplan-Meier (KM) estimation for automated metrics in section E.6. Moreover, the extension of the univariate CPH analysis to stage II CRC patient is available in section E.7.

6.4.6 Multivariate

For the multivariate analysis, we use forward variable selection. For each cohort, we select as potential variables the ones that achieve statistical significance in the univariate setting. We only retain the clinical variables common to all cohorts, namely age, gender, pT, pN, TNM, and microsatellite instable (MSI) status. To avoid high redundancy of the variables, we select for TSR and TBC estimation the variables that achieve the best fit across cohorts as TSR_{ROI1} and $\text{TBC}_{\text{RATIO}}$, respectively. In addition, we also keep stroma tissue distribution TD_{STR} as a variable. Other automated metrics are dropped. The results for OS and DFS survival are presented in Figure 6.17 for the aggregation of all cohorts with the computed IBS and C-Index scores.

In the first column, we highlight the clinical variable in the multivariate case. Out of the selected entries, age, gender, pT, and TNM are selected by the model for OS. We observe the same trend for the DFS. In columns two to four, we present the results where we add our proposed automated metrics as potential variables. If the designed metrics have significant statistical relevance, the model should select them during the forward selection. For the OS, the multivariate model keeps the distribution of stroma tissue at the boundary. For the DFS case, both TSR and TBC predictors are selected. In all cases, we notice that the depth of invasion disappears or is replaced by the automated approaches. It highlights that the pT variable is redundant when adding the automated predictions. The multivariate analysis restricted to the stage II cohorts is available in the supplementary section section E.7.

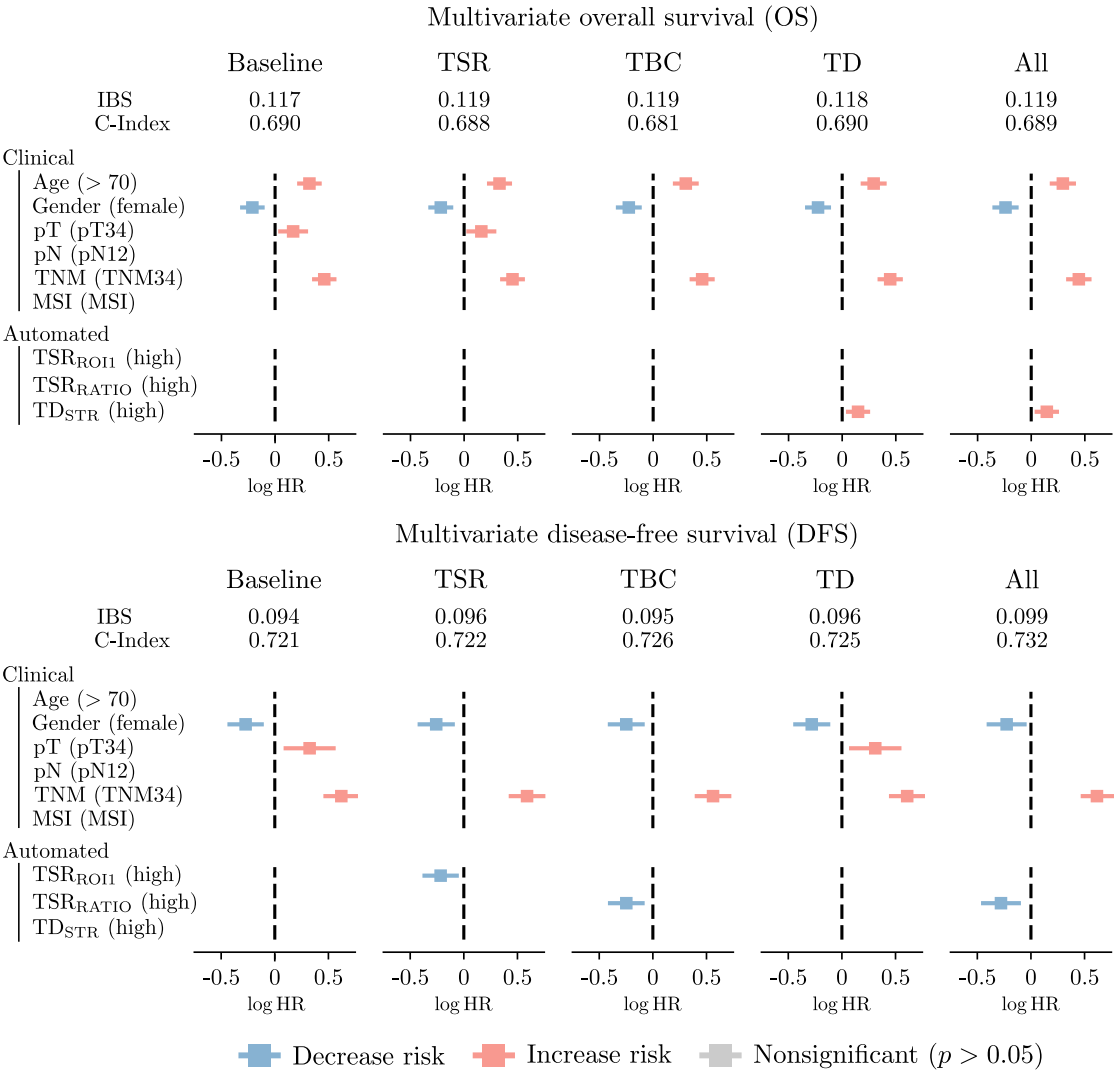


Figure 6.17 – Multivariate CPH estimation for OS based on clinical and automated metrics on cohorts aggregation. We consider a 5-year period for the study where features are binarized before model fitting. The selected group is indicated between parentheses. We report concordance index (C-Index) and integrated Brier score (IBS).

6.5 Conclusion

In this chapter, we focus on building clinically relevant metrics for survival analysis. More specifically, our work is centered on the estimation of TSR, TBC, and TMR, which are known to be interesting features for patient stratification. The assessment of all three metrics relies on manual annotation by expert pathologists. Such a task is tedious, prone to error, and acts as a break in the development of large cohort studies.

To tackle this problem, we propose to take advantage of our weakly supervised semantic segmentation (WSSS) C2R model to estimate clinical metrics in an automated fashion.

For TSR, we present multiple approaches that rely either on WSI-level statistics or ROIs identification. Regarding TBC, we first develop a TB locator and then introduce various estimates based on the computation of normal products, tissue ratio, and local interaction. The assessment of TMR is based on the ratio between the amount of detected tumor and mucin. Finally, we investigate the presence of STR tissue as the tumor boundary that we believe could have a high predictive power. To validate the quality of the presented metrics, we compare our automated predictions to expert annotations in terms of F_1-score and IOA.

We further explore the predictive power of the metrics by running univariate and multivariate analyses across five different cohorts. The statistics include more than 2,000 slides and 1,700 unique patients where OS and/or DFS are available. The survival analysis shows statistical significance for TSR, TBC, and stroma presence at the boundary. Even though TMR correlates with histological types, it does not, unfortunately, show up as a relevant metric based on the presented patient slides.

In a nutshell, the presented automated approach saves pathologists precious time while allowing large-scale studies. It highlights the predictive power of TSR, TBC, and stroma distribution regarding patient stratification. We show that our method has the potential for automated TSR assessment to be included in standard reporting. Moreover, complementary analysis on stage II cohort data shows encouraging values for DFS.

Throughout the development of our approach, we faced some limitations that could hinder the performance of our predictions. Firstly, the orientation of the tumor depends on the presence of muscle and adipose layer. Such layers are not always available and could lead to a performance drop. A solution to recover part of the detection would be to include the localization of the normal mucosa or muscularis mucosa. Secondly and lastly, the assessment of TBC and TSR is usually performed on multiple slides and aggregated at the patient level by taking the average or minimum. In our study, we are often limited to a single WSI per patient, which can lower the IOA. In future studies, we should investigate the effect of slide availability at the patient level. Moreover, an alternative to computing the TBC would be to merge the metrics into a single descriptor.

7 Conclusions

> Sais-tu seulement à quel point tu ne sais pas?
>
> *Batterie faible, Autotune*
> William Kalubi Mwamba

This chapter concludes the thesis. We first summarize the work and contributions in section 7.1. Then, we discuss the faced limitations and future directions in section 7.2.

7.1 Summary

In chapter 2, we introduce the main concepts of the thesis. We give the reader all the tools to properly understand the challenges linked to self-supervision in histopathology and colorectal cancer (CRC).

In chapter 3, we present our first contribution with the work of Divide-and-Rule (DNR). We highlight the advantages of combining self-supervised learning (SSL) and whole slide images (WSIs) structured data to learn tissue representation. We show that using staining information from hematoxylin and eosin (HE) to reconstruct original RGB images (HE to RGB) yields better feature representation compared to the traditional setting (RGB to RGB). In addition, we prove that the combination of spatial and feature proximity losses is critical to learning coherent tissue features. Finally, we demonstrate the prediction capability of our model by aggregating tissue representation at the patient level and performing survival analysis. The results highlight multiple sets of features that are relevant for patient stratification. Such features include tumor-to-stroma interactions as well as dense tumor areas.

Next, we move to chapter 4, where we introduce our second contribution as Self-Rule

to Multi Adapt (SRMA). With this work, we tackle the problem of WSIs domain gap when working with multi-source data. Through extensive experiments, we show that our model can benefit from multiple source data if available. It allows us to take advantage of the weakly-labeled data from the source site without asking expert pathologists for additional annotations. Moreover, we demonstrate that our model can handle previously unseen classes using an easy to hard (E2H) approach. It comes in handy as publicly available data often provide labels for a handful of categories, which might not correctly represent the complexity of our target data.

In chapter 5, we summarize our third contribution as coarse to refined (C2R) which aims to refine tissue segmentation. The previous approaches used a sliding window approach to perform tissue identification, which tends to produce coarse class representations. In this chapter, we prove that the combination of weakly supervised semantic segmentation (WSSS) and SSL can help the model to learn coherent segmentation maps. Moreover, we demonstrate that our approach can be trained using data from lazy annotators, which removes the need for pixel-wise annotations.

Finally, we present our fourth and last contribution in chapter 6. We take advantage of our previous model to automatically predict well-established metrics. We show that the automated predictions align with experts' annotations on multiple cohorts and thus can be used for extensive studies. Consequently, we highlight the correlation of the predicted metrics with various clinical variables over large patient sets. Moreover, through univariate and multivariate analysis, we show that tumor to stroma ratio (TSR), tumor border configuration (TBC), and tissue distribution can be used along with clinical metrics to stratify patient groups and, possibly, target better treatment. In this chapter, we demonstrate the applicability of our research to diagnostic routines.

Throughout this thesis, we highlighted the importance of open research. Sharing work and results is critical for reproducibility and developing novel approaches. Consequently, we make our research available online.

7.2 Limitations and Future Works

We now continue our chapter by discussing the limitations of our work and proposing future development directions. We first concentrate on two topics associated with data self-supervision and then discuss two additional issues related to clinical variables.

Structure Representation, the Never-ending Story: One of the significant limitations we face in this work is the creation of fine-grained segmentation. The problem is addressed in chapter 5 using WSSS and shows promising results. Still, we could observe a few limitations during the validation phase. For instance, even though the output resolution of the model has been improved, it remains insufficient to properly detect

small cell structures as tumor buds. The model would benefit from implementing tissue consistency terms to improve its segmentation prediction. For example, the model could impose local class uniformity during segmentation using superpixels [107, 154].

In addition, the rise of foundation models that include famous architectures such as BERT [44], DALLE [119], or GPT [18] should be investigated. Foundation models are defined as "any model that is trained on broad data (generally using self-supervision at scale) that can be adapted to a wide range of downstream tasks" [15]. What makes them different from the presented self-supervised approaches is their scale (*e.g.* 175 billion for GPT-3 whereas 85 million for DINO). They can be used for many tasks, including text synthesis, image description, and segmentation. They are already used in digital pathology to synthesize information from biomedical texts and histopathological images [94, 69].

Benefiting from Curated Data: The evolution of computational pathology is conditioned by the current state of the art (SOTA) in SSL. We can see the most recent works [30, 51] use the latest SSL approaches as DINO [23] and iBOT [163] to learn tissue embedding. However, a recent benchmark [75] suggests that this race to better SSL models does not highlight any clear winners as all approaches tend to perform reasonably well. The authors recommend putting more attention on creating large-scale curated domain-aligned datasets. This is confirmed in DINOv2 [111] where curated data significantly improve the embedding representation. In our work, the creation of in-house datasets is either performed by randomly sampling from WSIs (chapter 3 and chapter 4) or based on pseudo labels (chapter 5). As a result, the generated data can hardly be considered curated.

For future development, we encourage people to keep track of the current research in SSL. But more importantly, we recommend creating large sets of curated data to train SSL models. It could be achieved by building a simple image retrieval network based on cosine-similarity [111, 114] or ranking scores [132]. Moreover, as histological data are widely available, the resulting sets of data could easily include millions of relevant examples.

Gathering Information from Multiple Slides: Another critical aspect of this thesis is the availability of patient data and, more precisely, WSIs. For most cohorts, we have access to a single WSI per patient. However, more than one slide is needed to achieve proper statistics. Tumors are complex mediums whose structure might diverge based on different cuts. Moreover, our presented approaches to compute clinical metrics rely on the presence of muscle and adipose tissue to identify tumor orientation and progression. If the mentioned tissues are unavailable on the slide, the case is excluded from further analysis. These limitations can be observed in section 6.4 where the interobserver agreement (IOA) between the automated approach and pathologists for TBC varies a lot based on the selected slides.

Tumor representation will benefit from having access to multiple WSIs. Based on this conclusion, we encourage using additional WSIs data per patient in future work. In addition, the question of: "*how to properly aggregate the results given multiple patient slides*?" should also be investigated. It will help determine whether the tumor structure information is equally spread across cuts or is concentrated at specific locations, such as most invasive tumor parts. This aspect is critical to the building of clinically relevant metrics.

Clinical Metrics and Survival Analysis: The provided solution for metric assessment relies on a two-step approach. We first use machine learning models to infer WSI segmentation maps and then run our estimate of TSR and TBC based on the predicted outputs. The estimation of the metrics is based on hand-crafted features that try to mimic the pathologists' evaluation process. In future work, we would like to remove the need for hand-crafted features and intermediary processing by building an end-to-end network that predicts metrics directly from WSIs.

This also raises the interest in other applications that can benefit from this end-to-end approach, such as optimizing time-to-event (TTE) for survival analysis [86]. The task is challenging since survival data are usually scarce, thus making the training of end-to-end architectures tricky. A possible development idea would be to take advantage of the multiple-slide setting. Given several slides from the same patient, we can assume they all originated from distinct cases sharing the same TTE.

Last but not least is the use of genetic data. It allows us to investigate further the correlation of our prediction with other variables, such as microsatellite instable (MSI) (high/low), that are associated with CRC [89].

A Background - Supplementary Material

A.1 CRCTP Anomalies

During this thesis, we discovered multiple discrepancies between data labels and images in the colorectal cancer tissue phenotype (CRCTP) dataset [71]. To assess the quality of the dataset labels, we use a pre-trained architecture [64] to compute patch embeddings. We denote as $\mathbf{u}_i, \mathbf{v}_j \in \mathbb{R}^D$, the i-th and j-th embedding of dimension D of an image from the training set and testing set, respectively. We use the uniform manifold approximation and projection (UMAP) [100] to project the high-dimension embedding to a two-dimensional space. The procedure is similar to the t-distributed stochastic neighbor embedding (t-SNE) [140] except for the fact that the transformation learned by UMAP can be applied to new samples. As a result, we use a subset of our data (from train and test) to learn the UMAP transformation and then apply it to the whole dataset. This approach is less greedy regarding computational resources when working with big datasets.

The feature representations of both the training and testing set are depicted in Figure A.1. As the representation of the UMAP is learned with samples from the test and training set, we have a direct correspondence between features. At first sight, it seems that class distributions match. However, when looking closely, we identify areas with label discrepancies. To help visualize those errors, we fit a k-nearest neighbors (KNN) model to the projected training set data using $K = 9$ neighbors and then apply it to the test data to check for prediction errors in the UMAP space. We display the error density over the feature map. The darker the area, the denser the classification error between the two sets. These results are qualitative as the distances between features in the UMAP space are not Euclidean. Still, we report potential labeling errors between classes complex stroma (CSTR) and tumor (TUM), as well as stroma (STR) and muscle (MUS).

We further investigate the overlapping of the sets. For each sample of the training set, we look for the one in the testing set that maximizes similarity as the dot product $\text{sim}(\mathbf{u}_i, \mathbf{v}_j) = \arg\max_j \mathbf{u}_i^\top \mathbf{v}_j$. We found out a significant part of the results have queries

(a) Train embedding (b) Test embedding (c) Error density

Figure A.1 – Visualization of CRCTP feature embeddings. (a-b) Projection of the embedding space using t-SNE for train and test set in CRCTP. (c) classification error density between the two sets using a simple KNN classification.

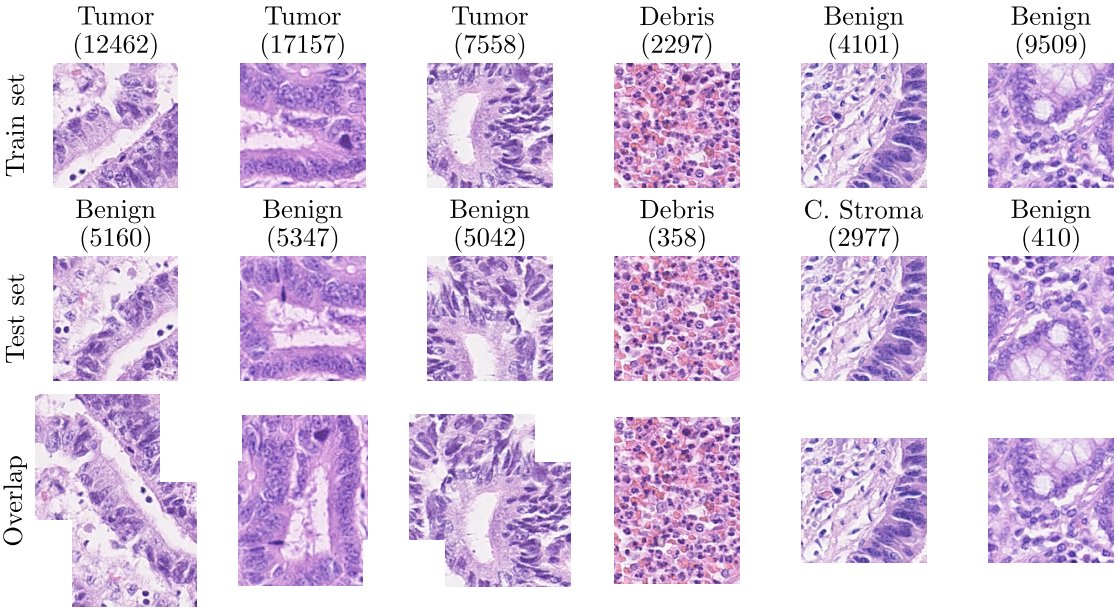


Figure A.2 – A non-exhaustive list of overlapping tiles between train and test set in CRCTP fold2 data. The fold does not ensure patient slides' and tiles' independence between the two sets. For each example, we report the unique id of the tile as well as their overlap.

in the test set that visually overlap. In the data description, the author mentioned that the data from the 2nd fold are split patient-wise, meaning that data from one slide cannot belong to both the training and testing set. In Figure A.2, we display a non-exhaustive list of overlapping tiles between the training and testing sets. Not only do part of the data from both sets belong to the same patient slides, but they also highly overlap (*e.g.* ID 9509 in train and ID 410 in test set achieve 100% overlap). Moreover, the labels are not consistent between the sets. Tiles that appear with label TUM (*e.g.* ID 12462, 17157,

7558) in training set end up as benign in the test set (*e.g.* ID 5160, 5347, 5042).

The presented results show evident discrepancies between the training and testing set. Moreover, we observe a redundancy of some tiles between the training and test sets. After reviewing with an expert, the training set labels "better" represent the actual tissue images. The data from the CRCTP still represent a helpful resource for strategies where no labels are required, such as unsupervised domain adaptation (UDA) or self-supervised learning (SSL).

A.2 Datasets Correspondence

In Figure A.3, we visually represent the classes for the main datasets used in this work. For the SemiCol data, we display cropped areas of the main images to allow a fair comparison with other datasets. For the in-house data, we present image samples from patient set $\mathcal{P}_A$. We add a reference value for the corresponding size of the tiles in pixels. The reference size is common to all tiles from the same dataset. Finally, if the name of the class differs from our definition, we state its original name at the top right of the tile.

A.3 Additional Cohorts Information

In this section, we report additional information about the available clinical variables. In Table A.1 and A.2, we show the details about the median age (with interquartile range), tumor location, tumor grade, lymphatic invasion, venous invasion, budding, microsatellite status, consensus molecular subtypes (CMS), and post-operative treatment as well as their definitions. We label as not available (NA) the entries with incomplete measures. For microsatellite instable (MSI), the mismatch repair status is defined based on immunohistochemistry (IHC) staining [6]. As a result, we do not have access to MSI-low and MSI-high status.

In Table A.3, we restrict the definition of the cohort to stage II colorectal cancer (CRC). We report the same variables as the general case presented in the main document.

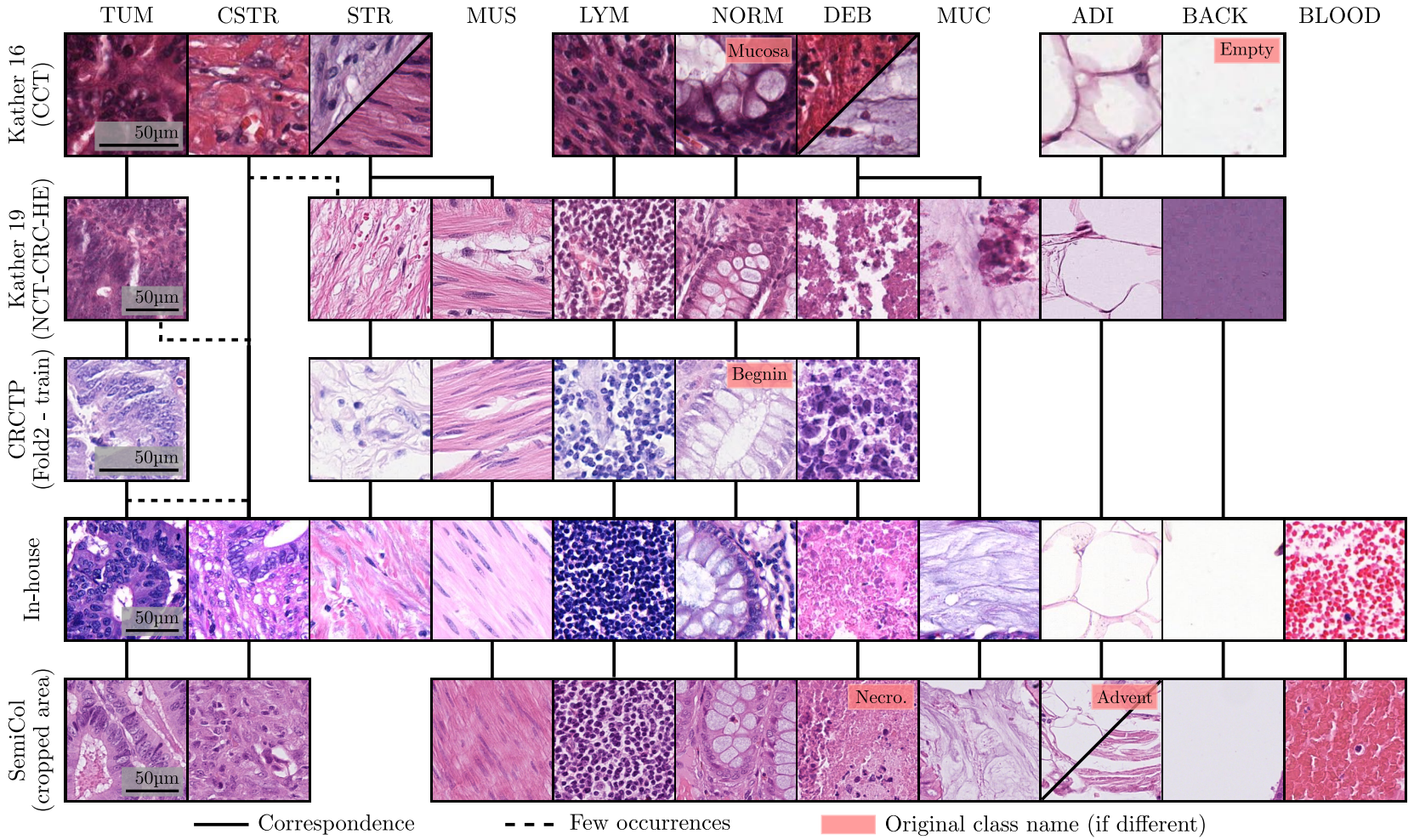


Figure A.3 – Tissue correspondence between main classification datasets Kather 16 (K16), Kather 19 (K19), CRCTP, In-House, and SemiCol. We display the original class name if it differs from the column name.

Table A.1 – Patient cohorts extended clinical variables.

Characteristics	Bern ($\mathcal{P}_A$)	Bern MSI ($\mathcal{P}_B$)	Nijmegen ($\mathcal{P}_C$)	TCGA ($\mathcal{P}_D$)	Toronto ($\mathcal{P}_E$)	All ($\mathcal{P}_{A-E}$)
Age (median [IQR])	71.6 [61.9 - 78.7]	74.0 [65.3 - 80.0]	70.0 [61.0 - 78.0]	67.0 [57.0 - 75.0]	69.0 [59.0 - 77.8]	70.0 [60.0 - 78.0]
Tumor site (%)						
Right	140 (38.4%)	104 (59.8%)	331 (59.9%)	167 (43.6%)	55 (47.4%)	797 (50.1%)
Left	167 (45.8%)	62 (38.7%)	222 (40.1%)	158 (41.3%)	61 (52.6%)	670 (42.1%)
Rectal	58 (20.9%)	8 (4.6%)	-	58 (15.1%)	-	124 (7.8%)
Tumor grade (%)						
g1	21 (5.5%)	16 (9.9%)	16 (2.9%)	NA	110 (93.2%)	163 (13.5%)
g2	287 (75.5%)	94 (58.0%)	391 (71.7%)	NA	8 (6.8%)	780 (64.7%)
g3	72 (18.9%)	52 (32.1%)	138 (25.3%)	NA	-	262 (21.7%)
Lymphatic invasion (%)						
l0	156 (43.7%)	85 (48.9%)	301 (69.8%)	NA	NA	542 (56.3%)
l1	201 (56.3%)	89 (51.1%)	130 (30.2%)	NA	NA	420 (43.7%)
Venous invasion (%)						
v0	219 (61.2%)	97 (55.7%)	NA	NA	96 (87.3%)	412 (64.2%)
v1	139 (38.8%)	77 (44.3%)	NA	NA	14 (12.7%)	230 (35.8%)
Budding (%)						
Low	107 (36.9%)	69 (55.2%)	NA	186 (72.4%)	27 (22.9%)	389 (49.2%)
Intermediate	88 (30.3%)	25 (20.0%)	NA	38 (14.8%)	31 (25.3%)	182 (23.0%)
High	95 (32.8%)	31 (24.8%)	NA	33 (12.8%)	60 (50.8%)	219 (27.7%)
MSI (%)						
MSS	257 (87.4%)	88 (50.6%)	376 (76.0%)	271 (68.8%)	88 (77.9%)	1080 (73.5%)
MSI	37 (12.6%)	86 (49.4%)	119 (24.0%)	123 (31.2%)	25 (22.1%)	390 (26.5%)
CMS (%)						
CMS1	NA	NA	NA	48 (12.8%)	NA	48 (12.8%)
CMS2	NA	NA	NA	166 (44.4%)	NA	166 (44.4%)
CMS3	NA	NA	NA	51 (13.6%)	NA	51 (13.6%)
CMS4	NA	NA	NA	109 (29.1%)	NA	109 (29.1%)
Postoperative therapy (%)						
No	126 (66.3%)	6 (60.0%)	435 (79.7%)	235 (59.0%)	NA	802 (70.1%)
Yes	64 (33.7%)	4 (40.0%)	111 (20.3%)	163 (41.0%)	NA	342 (29.9%)

Abbreviations: Not available or too few samples (NA). Interquartile range (IQR)

Table A.2 – Main definitions of clinical variables used in the main documents. For extended information, please refer to the reference publications [6, 95, 60].

Name	Definition
Tumor site	
Right	Locations include cecum, ascending, hepatic flexure, and transverse colon.
Left	Locations include splenic flexure, descending, sigmoid, and rectosigmoid junction.
Rectal	Location solely include rectum.
Histopathologic type	
Adenocarcinoma	Cancer, which forms in glandular epithelial cells (most common).
Mucinous	Characterized by the presence of extracellular mucin that accounts for at least 50% of the tumor volume.
Other	Less common CRC (*e.g.* signet ring cell carcinoma).
Tumor grade	
g1	Well differentiated, low grade (like healthy cells).
g2	Moderately differentiated, intermediate grade (somewhat like healthy cells).
g3	Poorly differentiated, high grade (less like healthy cells).
Lymphatic invasion	
l0	No.
l1	Yes, cancer cells within lymph vessels.
Venous invasion	
v0	No.
v1	Yes, cancer cells within blood vessels.
Budding [95]	
Low	0 to 4 tumor buds within 0.785 mm^2.
Intermediate	5 to 9 tumor buds within 0.785 mm^2.
High	more than 10 tumor buds within 0.785 mm^2.
MSI [6]	
MSS	Microsatellite stable, also referred to as mismatch repair proficient (MMR-p).
MSI	Microsatellite instable, also referred to as mismatch repair deficient (MMR-d). The distinction between MSI-L and MSI-H is performed through immunohistochemistry.
CMS [60]	
CMS1-4	Classification based on various factors such as the presence of specific mutations.
Postoperative therapy	
No	No.
Yes	Indicates that the patient received postoperative radiotherapy or chemotherapy.

Table A.3 – Patient cohorts with main clinical variables restricted to stage II CRC.

Characteristics	Bern ($\mathcal{P}_A$)	Bern MSI ($\mathcal{P}_B$)	Nijmegen ($\mathcal{P}_C$)	TCGA ($\mathcal{P}_D$)	Toronto ($\mathcal{P}_E$)	All ($\mathcal{P}_{A-E}$)
Patients	134	79	241	170	118	742
Slides	261	79	241	170	118	869
Sex (%)						
Male	77 (57.5%)	47 (59.5%)	116 (48.1%)	86 (50.6%)	68 (57.6%)	370 (51.1%)
Female	57 (42.5%)	32 (40.5%)	125 (51.9%)	84 (49.4%)	50 (42.2%)	362 (48.9%)
T-stage (%)						
T1	-	-	-	-	-	-
T2	-	-	-	-	-	-
T3	104 (77.6%)	48 (60.8%)	198 (82.2%)	159 (93.5%)	101 (86.3%)	610 (82.3%)
T4	30 (22.4%)	31 (39.2%)	43 (17.8%)	11 (6.5%)	16 (13.7%)	131 (17.7%)
N-stage (%)						
N0	134 (100%)	79 (100%)	241 (100%)	170 (100%)	118 (100%)	742 (100%)
N1	-	-	-	-	-	-
N2	-	-	-	-	-	-
TNM (%)						
I	-	-	-	-	-	-
II	134 (100%)	79 (100%)	241 (100%)	170 (100%)	118 (100%)	742 (100%)
III	-	-	-	-	-	-
IV	-	-	-	-	-	-
Histopathologic type (%)						
Adenocarcinoma	109 (82.6%)	57 (75.0%)	174 (72.5%)	136 (83.4%)	NA	476 (77.9%)
Mucinous	23 (17.4%)	19 (25.0%)	66 (27.5%)	27 (16.6%)	NA	135 (22.1%)
OS (%)						
Alive	107 (79.9%)	27 (77.1%)	199 (82.6%)	152 (89.4%)	101 (85.6%)	586 (84.0%)
Dead	27 (20.1%)	8 (22.9%)	42 (17.4%)	18 (10.6%)	17 (14.4%)	112 (16.0%)
DFS (%)						
Free	109 (94.0%)	13 (92.9%)	220 (92.1%)	NA	98 (83.8%)	440 (90.5%)
Recurrence	7 (6.0%)	1 (7.1%)	19 (7.9%)	NA	19 (16.2%)	46 (9.5%)
5-year OS	92.1%	78.2%	89.1%	62.1%	84.8%	90.5%
5-year DFS	NA	NA	91.2%	NA	82.8%	92.5%

Abbreviations: Not available or too few samples (NA).

B DNR - Supplementary Material

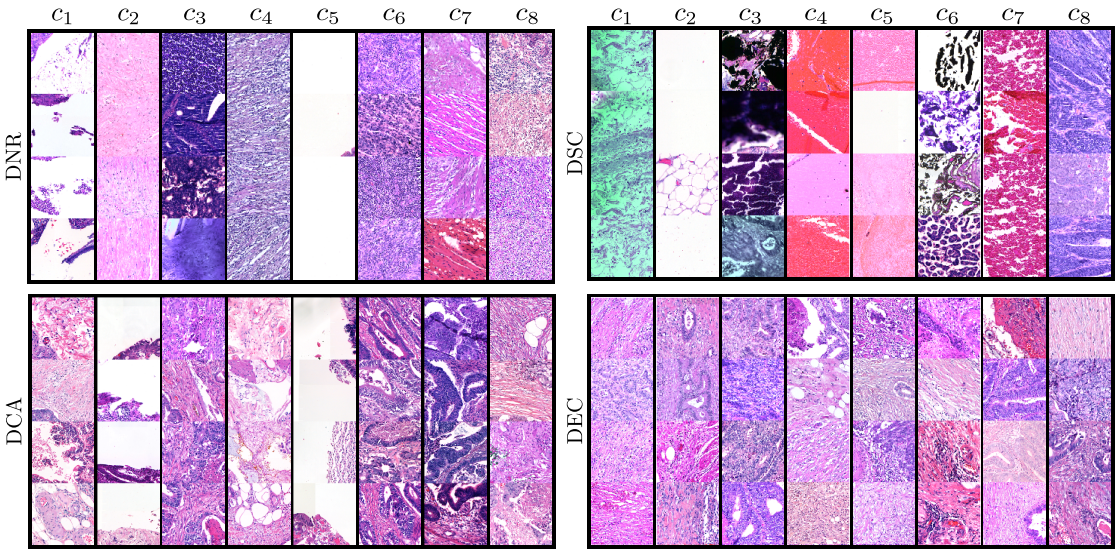


Figure B.1 – Top cluster elements based on $K = 8$ for Divide-and-Rule (DNR) (ours), spatial consistency (DSC), clustering assignment (DCA), and embedding clustering (DEC). The patches are samples from cohort $\mathcal{P}_A$.

B.1 Spherical Clustering

This section presents the extended result for the spherical K-means (SPKM) clustering. We display in Figure B.1 and Figure B.2 the results for the $K = 8$ and $K = 16$ cases, respectively. The clustering is applied to the feature representation of the Divide-and-Rule (DNR) (ours), spatial consistency (DSC), clustering assignment (DCA), and embedding clustering (DEC).

For the DSC, we observe the model is biased toward color representation. The cluster c_1 ($k = 8$) and c_{11} ($k = 16$) highlight patches that have a high green component (*i.e.*

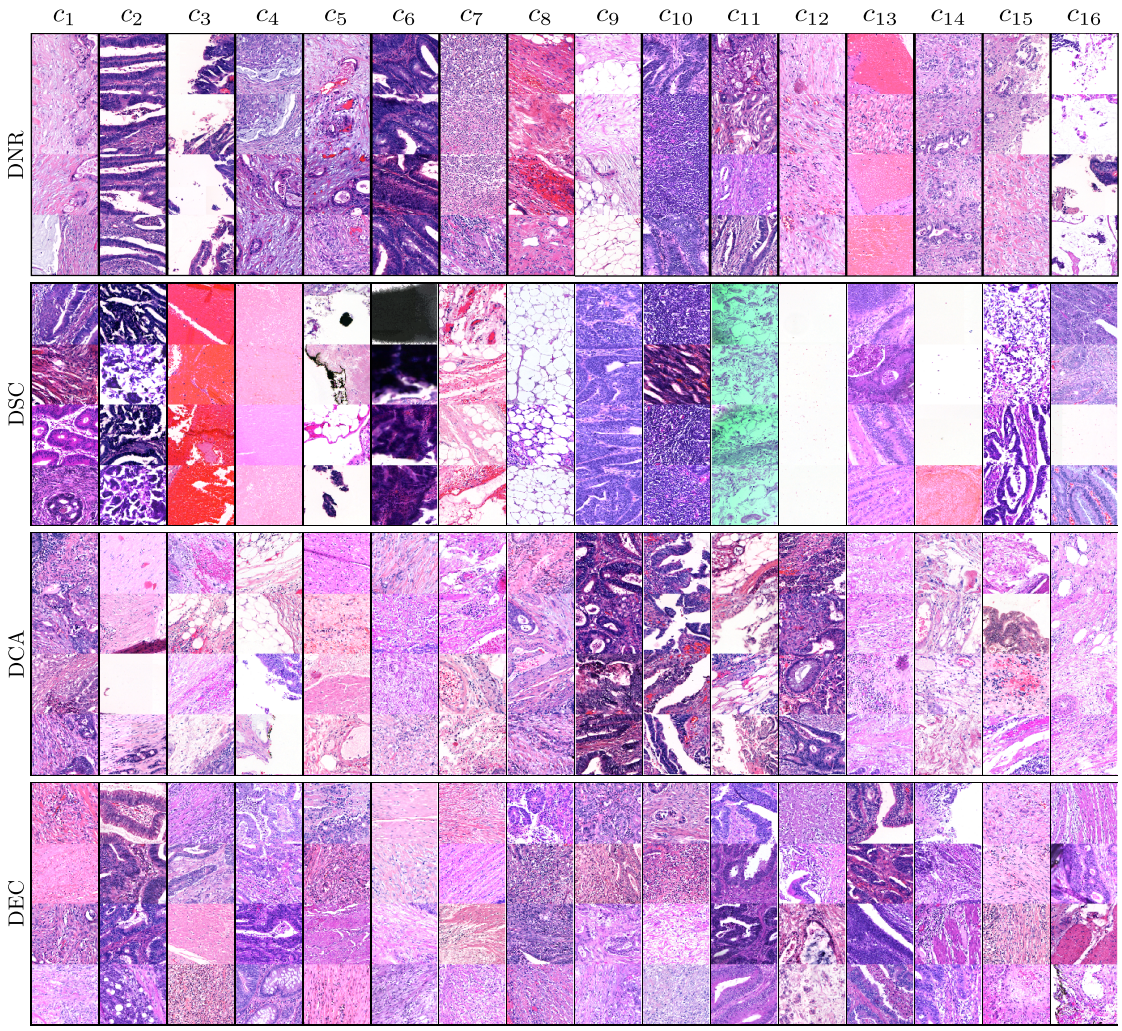


Figure B.2 – Top cluster elements based on $K = 16$ for Divide-and-Rule (DNR) (ours), spatial consistency (DSC), clustering assignment (DCA), and embedding clustering (DEC). The patches are samples from cohort $\mathcal{P}_A$.

pen marks on the whole slide image (WSI)). Such features are not relevant for survival analysis. For the DCA, DEC, and DNR, the clusters present more diverse information. At this stage, it is difficult to assess the relevance of the cluster information.

B.2 Reconstruction

This section shows the qualitative reconstruction of the decoder. The results are depicted in Figure B.3. The model takes as input a patch (top row). The hematoxylin and eosin channels are then extracted from the input patch and fed to an encoder. Finally, the decoder reconstructs the output image based on the feature representation (bottom row).

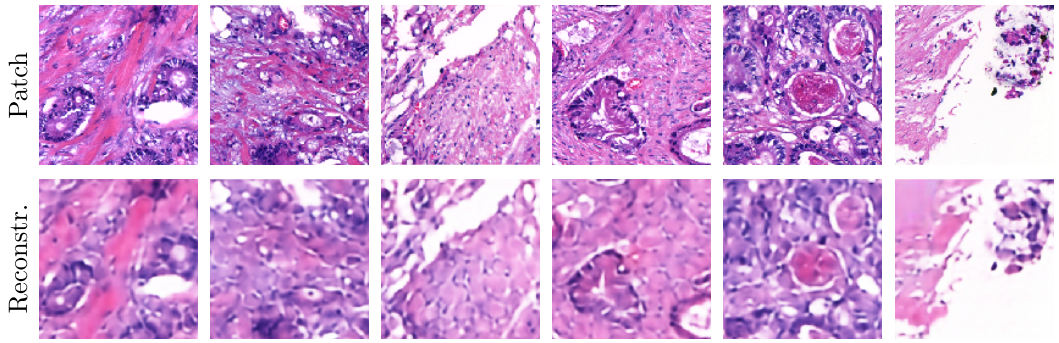


Figure B.3 – Image reconstruction for DNR. The model use hematoxylin and eosin (HE) channel to retrieve RGB source images.

We can see that the reconstruction of the image is by far not optimal. It mainly appears as a blurred image. This is because the decoder is composed of a succession of interpolation layers. Moreover, the decoder is designed as a shallow network, so the encoder captures most of the information.

C SRMA - Supplementary Material

C.1 Selection of Self-supervised Model

To assess which self-supervised model is more fitting our UDA setting, we compare the performances of several state of the art (SOTA) self-supervised methods (SimCLR [31], SupContrast [81], and MoCoV2 [32]), as well as the performance of the standard supervised learning approach when facing different levels of data availability. The results are presented in Table C.1. We report the performance of the single domain classification on K16 and K19. The supervised approach uses ImageNet pre-trained weights. The self-supervised baselines are trained from scratch. After self-supervised training, we freeze the weights, add a linear classifier on top, and train it until convergence. For SupContrast [81], we jointly train the representation and the classification as described in the original paper.

MoCoV2 [32] outperforms the two other SOTA approaches. On K16, the model gains up to 10% in terms of the F_1-score to the other self-supervised baselines. In addition, MoCoV2 gives competitive results with the supervised baseline that is initialized with ImageNet weights. It shows that MoCoV2 is able to efficiently learn from unlabeled data and create a generalized feature space. It mainly comes from the combination of the momentum encoder and the access to many negative samples. Hence, we adapt MoCoV2 for our proposed UDA method.

C.2 Patch Classification - t-SNE Projection

In this section, we present the complementary results to the ones in subsection 4.2.2 for patch classification. The embeddings of all baselines and our proposed approach are displayed in Figure C.1 using t-SNE visualization. We show the alignment between the source (K19) and target (K16) embedding domain, as well as classes-wise.

Table C.1 – Classification results of the different SOTA self-supervised approaches, as well as the supervised baseline on the K19 and K16 patch classification tasks. We present the results for different percentages of available training data. The top results are highlighted in bold. We report the weighted F1 score.

	K16 Labels fraction			K19 Labels fraction		
Methods	10%	20%	50%	1%	2%	5%
Supervised‡	85.8^{**}	86.5^{**}	87.9^{**}	$\mathbf{89.2}^{+}$	$\mathbf{89.9}^{+}$	$\mathbf{90.5}^{+}$
SimCLR [31]	79.6^{**}	78.9^{**}	78.6^{**}	76.9^{**}	79.4^{**}	80.7^{**}
SupContrast [81]	60.8^{**}	73.2^{**}	80.8^{**}	78.7^{**}	81.6^{**}	85.0^{**}
MoCoV2 [32]	**88.5**	**90.2**	**91.1**	**89.9**	**90.3**	**90.6**

‡ Model initialized with ImageNet pre-trained weights.
$^{+}$ $p \geq 0.05$; * $p < 0.05$; ** $p < 0.001$; unpaired t-test with respect to the top result.

With the source only approach, we can observe the lack of domain alignment between the feature spaces. Here, the model learns two distinct distributions for each set. On the other side, our approach shows a satisfactory alignment of domains compared to most baselines. The target complex stroma (K16) is linked to tumor, debris, lymphocytes, and stroma in the source domain (K19).

C.3 Multi-source Dataset Sampling Ratio

When performing multi-source domain adaptation, we assume all the source and target samples are from the same distribution. When sampling from $\mathcal{D}$, we have an equal probability of drawing a sample from the source or the target domain. In this section, we analyze the importance of sapling the source and target domains during the pre-training stage. We use K19 and K16 as source datasets and CRCTP as the target dataset. For K19 and K16, only 1% and 10% of the source labels are used, respectively. The classification performance results on the CRCTP dataset are presented in Table C.2. We indicate the multi-source scenario ($1:1$ or $K:1$), the sampling probability for each of the datasets, and the batch size.

The cross-domain matching using the $K:1$ scenario shows the highest variance, and its performances can vary by up to 2.6%. Overall, we can observe that balanced probability between all sets, namely $\frac{1}{3}$ each, gives similar results across all multi-source scenarios. In addition, when lowering the sampling probability of K16 we can see a drop in performances. It suggests that it is essential to have a balanced sampling strategy even if one of the source sets (*i.e.* K16 with 5,000 examples) is much smaller.

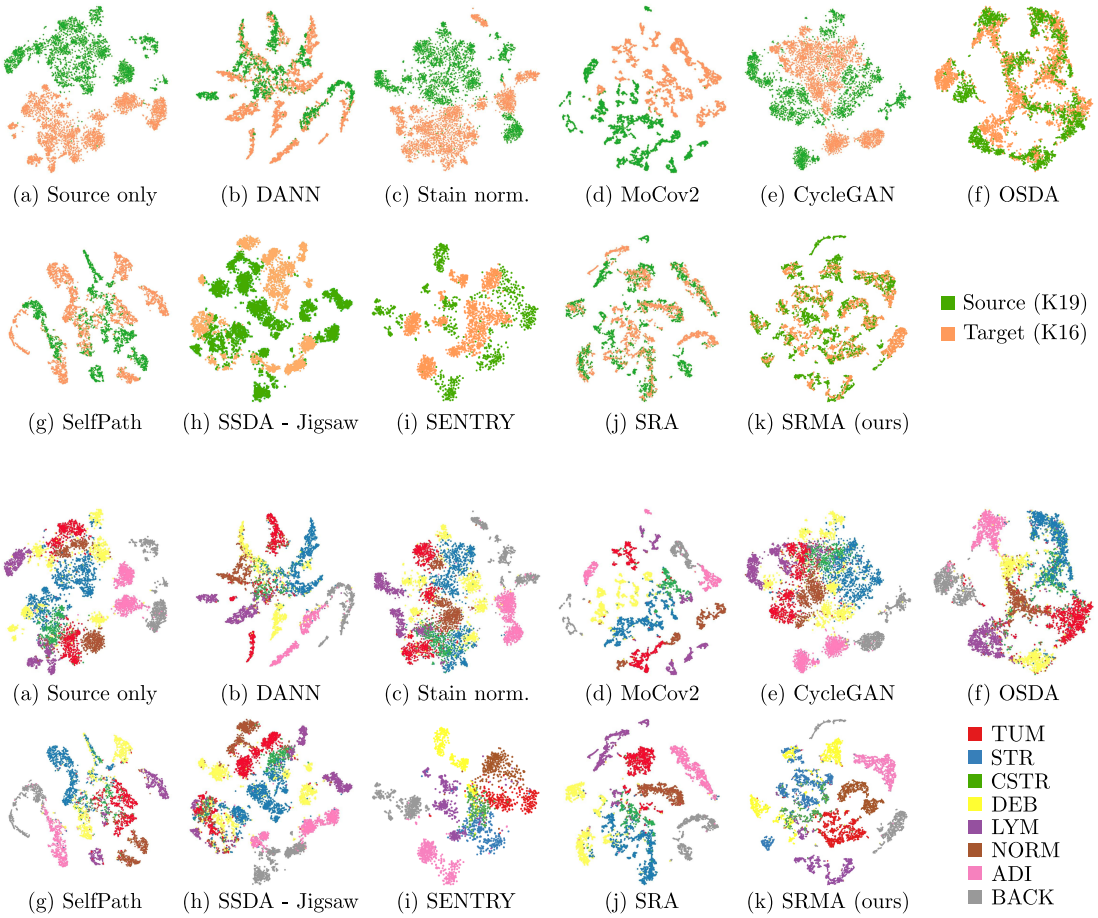


Figure C.1 – t-SNE projection of the source (K19) and target (K16) domain embeddings. We show the alignment of the embedding space as well as the individual classes for all presented models between the source and target domain. The classes of K19 are merged and relabeled according to the definitions in K16. The standard supervised approach is depicted in (a). We compare our approach (i) to other domain adaptation methods (b-j) as [54, 96, 27, 165, 124, 85, 153, 116], respectively. Our approach (k) qualitatively shows the best source and target domain alignment.

C.4 Multi-source - t-SNE Projection

Figure C.2 shows the visualization of the embedding for the proposed multi-source domain adaptation in subsection 4.2.7. It highlights the alignment of the feature space between the two source sets (K19, CRCTP) and our in-house dataset.

We observe that for each source domain, the categories are well clustered. Moreover, we notice that the classes shared by both domains (*i.e.* tumor, stroma, debris, lymphocytes, normal mucosa, and muscle) overlap. In addition, the domain-specific tissues (*i.e.* adipose, background, mucin, and complex stroma) form individual groups and are independent. Subsequently, our approach was able to properly correlate similar tissue definitions across

Table C.2 – Study of the multi-source domain performance of the Self-Rule to Multi Adapt (SRMA) approach with different sampling ratios. We use K19 and K16 as source datasets and CRCTP as the target dataset. We compare the introduced multi-source approaches defined in Equation 4.12 to Equation 4.15, where $1:1$ and $K:1$ refers to the one-to-one and K-to-one setting, respectively. We report the F1 score for the individual classes and weighted F_1 score (W-F_1) as the overall mean performance (all) averaged over 10 runs.

	Multi-source		Sampling ratio									
Model	$\mathcal{L}_{\text{IND}}$	$\mathcal{L}_{\text{CRD}}$	K19	K16	CRCTP	Batch size	TUM	STR‡	LYM	NORM	DEB	W-F_1
DeepAll [48]	-	-	-	-	-	128	72.4**	88.6**	43.6**	53.2**	71.8**	73.2**
SRA[1]	1 : 1	1 : 1	0.25	0.25	0.50	128	86.2**	87.6**	66.7**	71.0**	**80.5**	81.8**
SRMA	1 : 1	1 : 1	0.25	0.25	0.50	128	**92.5**	88.4**	68.7**	68.3**	74.2*	82.9*
SRMA	K : 1	1 : 1	0.25	0.25	0.50	128	91.5*	87.6**	**70.7**	**75.0**	65.7**	82.7*
SRMA	1 : 1	K : 1	0.25	0.25	0.50	128	90.1**	**90.1**	**69.6**$^{+}$	72.9**	71.6**	**83.6**
SRMA	K : 1	K : 1	0.25	0.25	0.50	128	**91.6**	87.4**	68.7**	73.9**	53.3**	81.2**
SRMA	1 : 1	1 : 1	0.33	0.33	0.33	128	**92.9**$^{+}$	87.8**	68.3**	65.3**	72.0*	82.0**
SRMA	K : 1	1 : 1	0.33	0.33	0.33	128	**93.1**	87.3**	70.5**	**78.3**	66.9**	83.4*
SRMA	1 : 1	K : 1	0.33	0.33	0.33	128	92.5*	**89.7**	**71.6**	73.0**	66.9**	**83.8**
SRMA	K : 1	K : 1	0.33	0.33	0.33	128	92.2*	88.6**	66.1**	74.3**	**74.5**	83.4*
SRMA	1 : 1	1 : 1	0.40	0.20	0.40	128	90.5**	88.3**	63.8**	71.8**	66.1**	81.5**
SRMA	K : 1	1 : 1	0.40	0.20	0.40	128	90.8**	**89.8**	62.0**	**74.7**	64.1**	82.2**
SRMA	1 : 1	K : 1	0.40	0.20	0.40	128	92.0*	88.6**	**69.5**	73.7**	64.8**	82.8**
SRMA	K : 1	K : 1	0.40	0.20	0.40	128	**92.7**	89.3**	65.8**	**74.7**$^{+}$	**75.2**	**83.8**

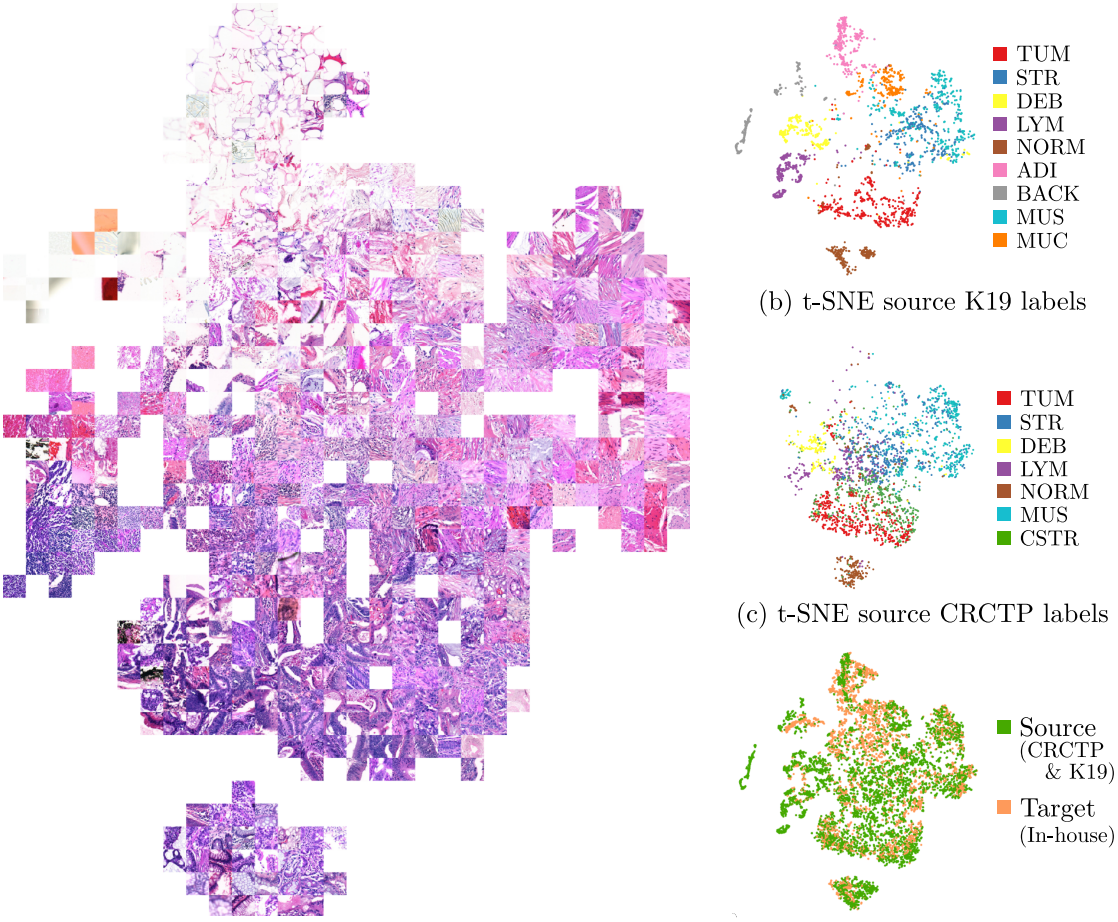
‡ The STR and MUS classes are merged as STR class; DEB and MUC classes as DEB.
$^{+}$ $p \geq 0.05$; * $p < 0.05$; ** $p < 0.001$; unpaired t-test with respect to top

the source domains while maintaining domain-specific tissue representation.

Looking at the source and target projection, we discern a batch of tissue (center-top) that does not align with the source domain. When associated with the patches visualization, we can recognize tiles with loose stroma, collagen, or blood vessel representation. Rightfully, none of the mentioned classes were present in the source domain, thus proving the usefulness of the easy-to-hard approach.

C.5 Patch-based Segmentation of WSIs from the TCGA Cohort

In this section, we highlight the performance of our framework on a publicly available WSI (UUDI: 2d961af6-9f08-4db7-92b2-52b2380cd022) from the the cancer genome atlas (TCGA) colon cohort [128, 129]. We apply our trained SRMA framework, as described in subsection 4.2.3, where K19 is used as the source domain and our in-house domain as the target one. We show the original image, classification output, and the tumor class probability map of our proposed SRMA method in Figure C.3.



(a) t-SNE projection of target patches

(b) t-SNE source K19 labels

(c) t-SNE source CRCTP labels

(d) t-SNE source and target

Figure C.2 – t-SNE visualization of the SRMA model trained on CRCTP, K19 and the in-house dataset. All sub-figures depict the same embedding. (a) Patch-based visualization of the embedding. (b-c) Distribution of the labeled source samples. (d) Relative alignment of the source and target domain samples.

The model is able to classify tissue across the whole slide accurately. Moreover, the pipeline gives a somewhat detailed output, which is a remarkable performance for a patch-based approach that is not explicitly designed for segmentation purposes. Furthermore, the model is agnostic to artifacts such as permanent marker spots (green marks on the bottom left). The tumor prediction map gives an overview of the tumor class probability across the WSI. This class is particularly interesting, as tumor detection is essential for many downstream tasks (*e.g.* detection of the invasive front or the tumor stroma ratio).

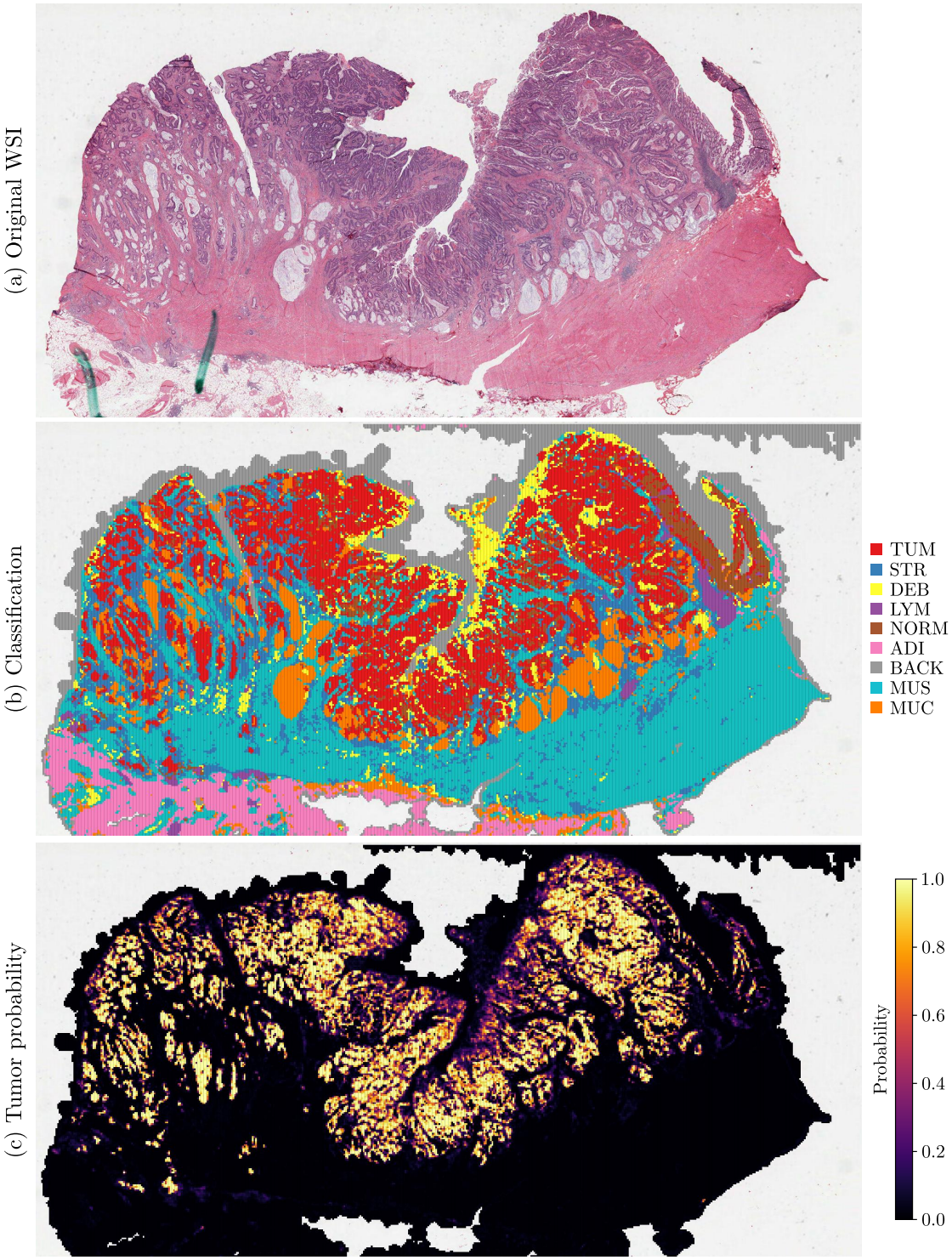


Figure C.3 – Segmentation results on a WSI from the TCGA cohort achieved by our SRMA model trained using K19 as the source dataset and our in-house set as the target dataset. We show the (a) original image, (b) classification output, and (c) tumor class probability map.

D C2R - Supplementary Material

D.1 Scanner Comparison - Tumor and Stroma

In Figure D.1, we show the output segmentation for tumor and stroma detection. The two rows depict the results for scanners A and B, respectively. We display the original HE image in the first column. Then, we show the generated annotation based on the consecutive cuts. Finally, we present the results for our SRMA and coarse to refined (C2R) approaches. For the classification, tissues that are not identified as either tumor or stroma are set to other (*i.e.* gray).

The prediction on scanner B achieves better performance. Regarding the methods, we observe the refinement of the output resolution for the C2R method. The architecture produces a fine-grained segmentation map. It is a considerable improvement with respect to SRMA.

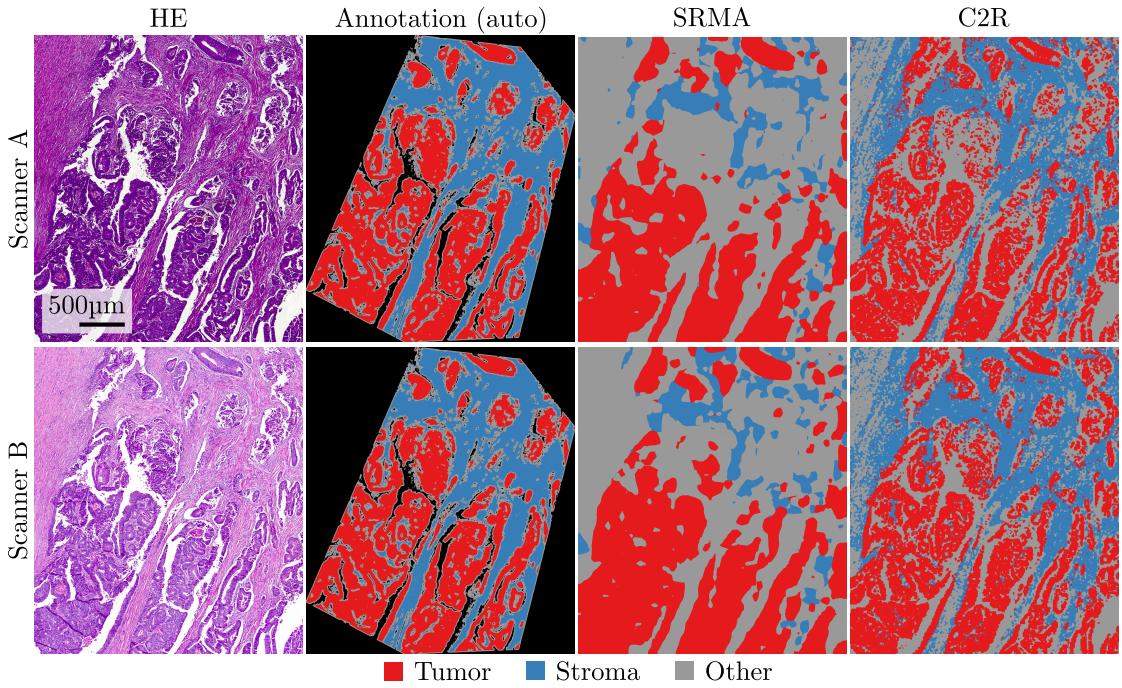


Figure D.1 – Scanner and method comparisons of tissue segmentation for tumor, stroma, and other detection. We display the original HE region, the generated annotations, and predictions for SRMA [2] and C2R. The rows show the difference in predictions scanner-wise.

E Building Clinically-Relevant Metrics - Supplementary Material

E.1 Tumor Area Estimation

When computing tumor border configuration (TBC), we must rely on a good tumor area estimation. The first estimation of the tumor area is made using the tumor channel from the segmentation algorithm. Unfortunately, this detection does not include debris and mucinous areas. In diagnosis, both sites are contained in the primary tumor area. A simple solution would be to merge the detection of the three channels as tumor, mucin, and debris to estimate the tumor's main area better. However, debris includes other artifacts across the WSIs that are not linked to the presence of the tumor. In addition, mucin is present in normal mucosae crypts, which is not correlated with the tumor area.

To overcome this issue, we use a region-growing approach. We use the tumor detection points as seeds for the algorithm. Note that the tumor is preprocessed to remove small detection points. The model then expands the area to include surrounding debris and mucinous tissue iteratively. Doing so ensures that all the selected tissues are directly connected to the tumor area. An overview is depicted in Figure E.1. We stop the expansion of the area when no the expanding area remains unchanged between two steps.

E.2 Region of Interest Estimation

In Figure E.2, we observe an example of the detection of the ROIs at the WSI level. We display the top $K = 3$ regions and the area selected by the expert pathologist for manual evaluation. The local estimation of TSR is reported along with the area's name. For the ROIs, we show the HE original WSI crop at 2.5×.

The presented example is labeled as a TSR-low case by the experts. It is validated by the evaluation of TSR within the annotated area by the algorithm (*i.e.* $\text{TSR}_{\text{ANNO}} = 0.180$). When looking in more detail at the segmentation map, we can see that the manually

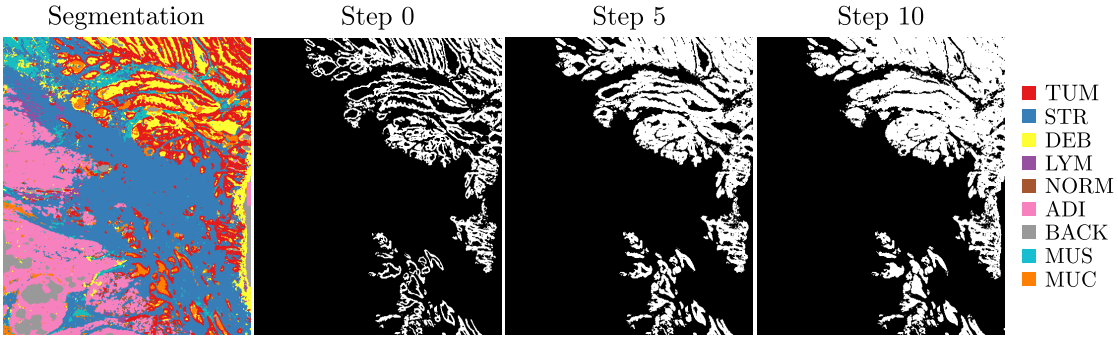


Figure E.1 – Iterative tumor estimation. We start the estimation from the filtered tumor detection (step 0). The model then iteratively adds neighbor tissues (debris and mucin) using morphology to expand the tumor area. The final estimation is given in step 10.

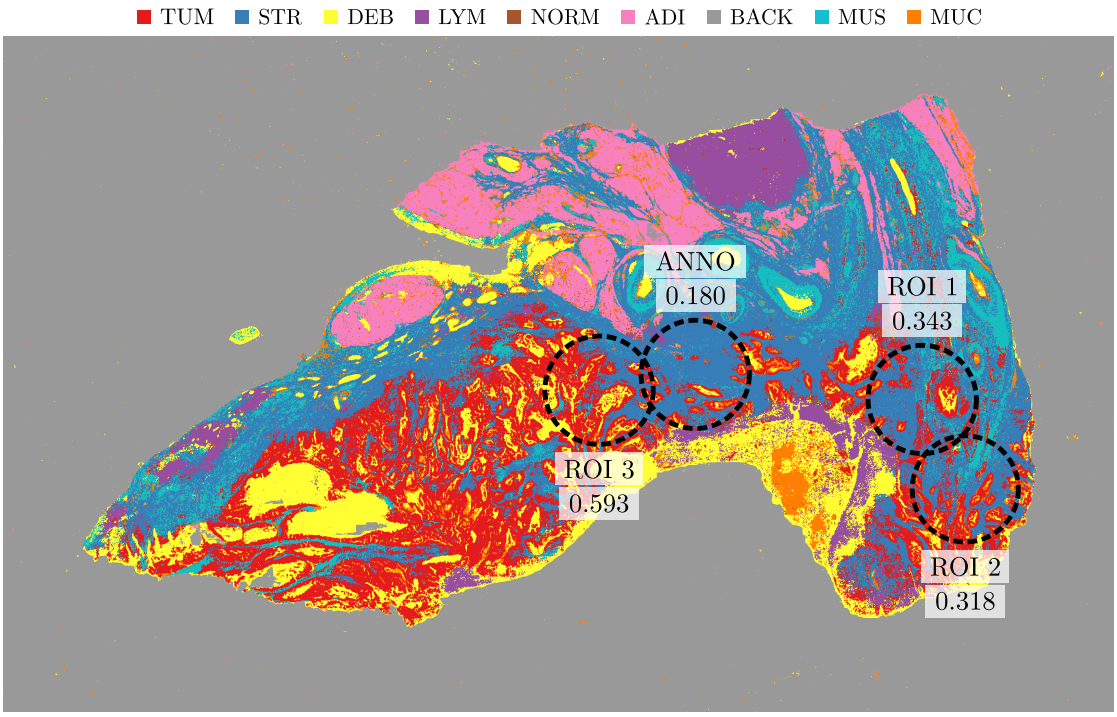
selected area does not seem to match the "four directions" rule. Here, the selected region includes tumor buds that are hard to detect by the segmentation algorithm. However, looking at the first ROI, we can observe a similar tumor structure as the annotated area, including sparse tumor blobs surrounded by stroma. When going through the WSIs, we often encounter cases where multiple locations fit the requirements for best ROI. We scarcely faced examples where only a single location of the WSI was determining the classification into TSR-low group. In other words, detecting the optimal ROI location is not limited to a single area.

The estimations of the TSR for the first two ROIs are similar (*i.e.* $\text{TSR}_{\text{ROI1}} = 0.342$ and $\text{TSR}_{\text{ROI2}} = 0.318$) and depict a TSR-low case based on a 50% cutoff. The third area, on the contrary, represents a TSR-high case (*i.e.* $\text{TSR}_{\text{ROI3}} = 0.593$. If we take the average of the K detected area, we still end up in a TSR-low case. However, here we see how sensitive the TSR group estimation is at the WSI level. Setting the number of ROIs higher would change the classification of the WSI toward a TSR-high case. Due to the large size of the selected lens, the value of K needs to be kept small.

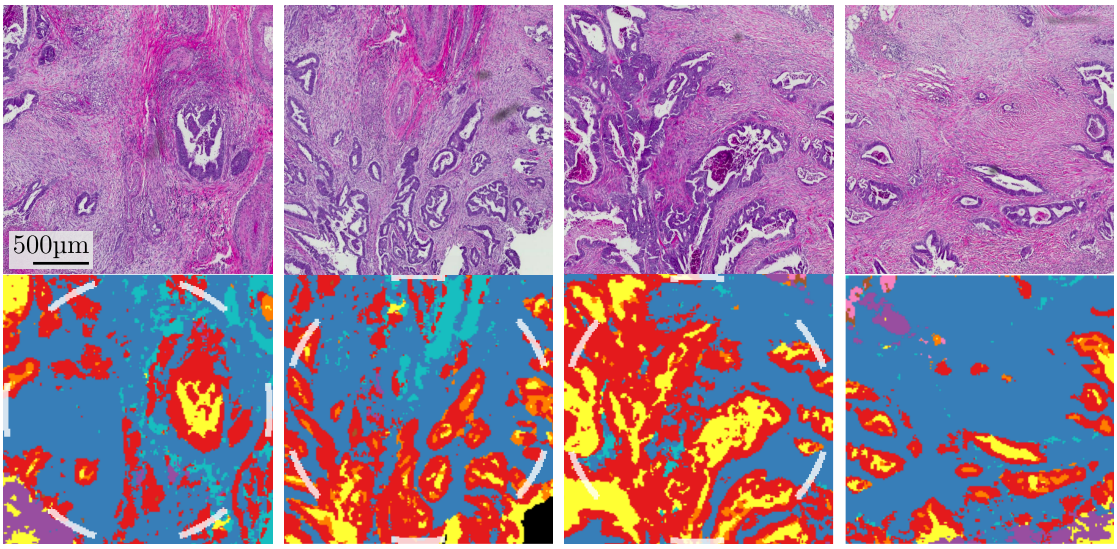
E.3 SRMA metric predictions

In this section, we elaborate on the difference in metric predictions between C2R and SRMA [2] models. The metrics are computed on cohort selected our TSR work [3] using SRMA.

In Figure E.3, we present the difference between the SRMA and C2R approach. The first two columns show the correspondence between metrics prediction. We observe that for both cohorts, there is a linear correlation with a Pearson score of 0.739 and 0.824. Still, the SRMA approach tends to predict higher TSR estimates. It is linked to the underestimation of the stroma content. Finally, in the last column, the differences at the WSI level are depicted.



(a) Segmentation of WSI and location of ROIs



(b) ROI 1 (c) ROI 2 (d) ROI 3 (e) Annotation

Figure E.2 – Visualization of the regions of interest (ROIs) localization for tumor to stroma ratio (TSR) estimation. (a) Segmentation of a reference WSI with localization of the top $K = 3$ ROIs and manual annotation. (b-d) Top K ROIs with their WSI local crop and detection filter (white). (e) Manual annotation by the expert pathologist.

Figure E.4 present the correlation between the TSR metrics automatic prediction and the pathologist annotations. The TSR correlate highly with the expert's annotations. Unfortunately, the estimates tend to predict high values. In this case, using a cutoff at

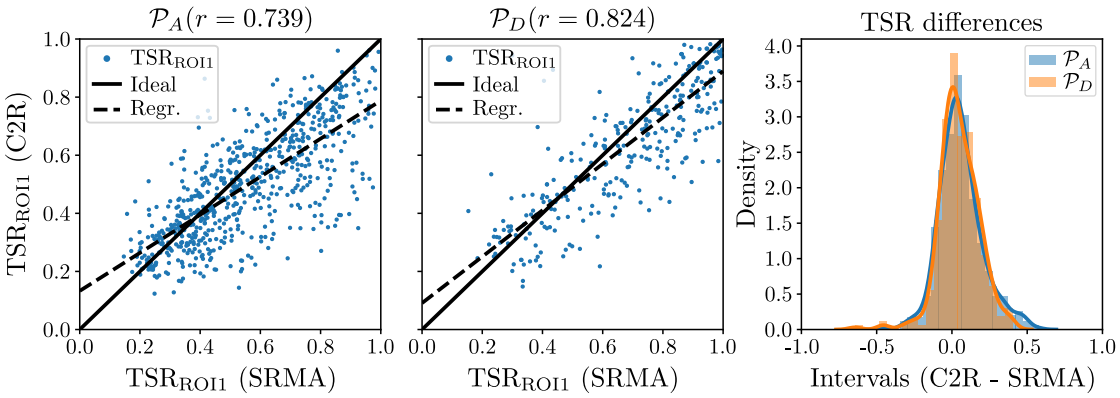


Figure E.3 – Comparison on TSR_{ROI1} prediction between SRMA and C2R models. (a-b) Correlation of predictions on $\mathcal{P}_A$ and $\mathcal{P}_D$ cohorts. The ideal one-to-one correspondence (ideal), linear regression (regr.), and Pearson correlation r are depicted. (c) Slide-level differences between SRMA and C2R models.

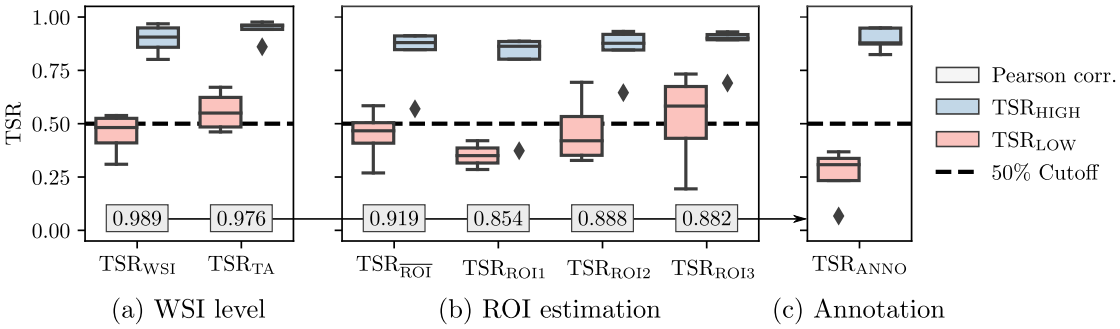


Figure E.4 – Comparison of TSR prediction based on SRMA model [2]. We use a 50% cutoff to split between TSR high (blue) and low (red). We report the Pearson correlation (gray) between automated predictions and the annotated area. (a) Estimation at the WSI level with and without tumor-associated stroma (TA-STR). (b) Detection of the top $K = 3$ ROI and averaged results. (c) TSR estimated on manually annotated ROIs.

50% is not optimal. A more reasonable threshold would lie around 75%. However, this does not fit with the TSR definition.

Finally, in Figure E.5, we compute the interobserver agreement (IOA) for TBC estimation using three different approaches on two cohorts. We observe a relatively low IOA, which suggests that the predictions do not align with expert annotations.

E.4 Correlation: TSR and Clinical

Here, we present the correlation of the TSR estimation with the clinical feature. We present the three main estimation as TSR_{WSI}, TSR_{ROI1}, and $\text{TSR}_{\overline{\text{ROI}}}$. Out of all the clinical features, we selected the most relevant ones as pT (depth of invasion of the

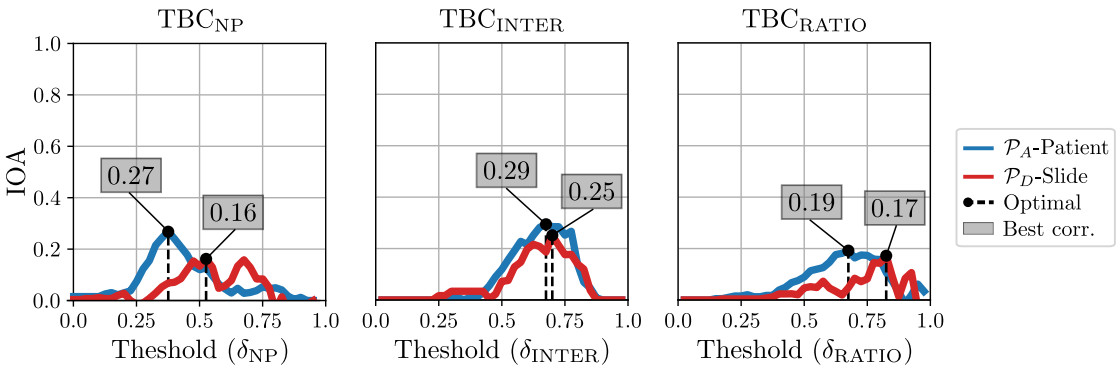


Figure E.5 – Evolution of interobserver agreement (IOA) between TBC estimations and expert's annotations as a function of the metric threshold. We display each cohort's optimal threshold and IOA for SRMA [2].

tumor), pN (number of lymph nodes affected), and the overall cancer stage (I to IV). The results are presented in Figure E.6 for five different cohorts.

We observe a similar trend for all TSR metrics. The prediction of the depth of invasion makes the best predictor. Lower TSR correlates with higher invasion depth, cancer stage, and lymphatic invasion. It is expected that cancer with high tumor-associated stroma tends to have worse survival outcomes than dense tumors. Note that we observe the same trend even for $\mathcal{P}_E$, which is limited to stage II patients. It indicates that TSR could be used to stratify stage II patients.

E.5 TBC - Examples

In Figure E.7, we highlight the performance of the TBC estimation on three different tumor areas (top row). The areas are handcrafted to represent three standard interfaces between tumor and stroma. The first example depicts a slowly moving border. Here, the edge is uniformly pushing through the tissue. The second example shows a "finger-like" growing pattern. The tumor grows using what can be identified as fingers. In the third and last case, we observe multiple small tumor structures ahead of the primary tumor area. These regions are composed of small tumor structures such as tumor buds. The last two mentioned patterns are connected to infiltrating cases.

We display the evolution of the local TBC for the metrics TBC$_{NP}$, TBC$_{RATIO}$, and TBC$_{INTER}$. We use a threshold for each metric to distinguish between the pushing and infiltrating aspect of the local TBC. The value of the thresholds are selected for displaying purpose. The overall TBC is estimated along the tumor border (TB) and represented as the percentage of TB location where the local TBC is higher than the selected threshold.

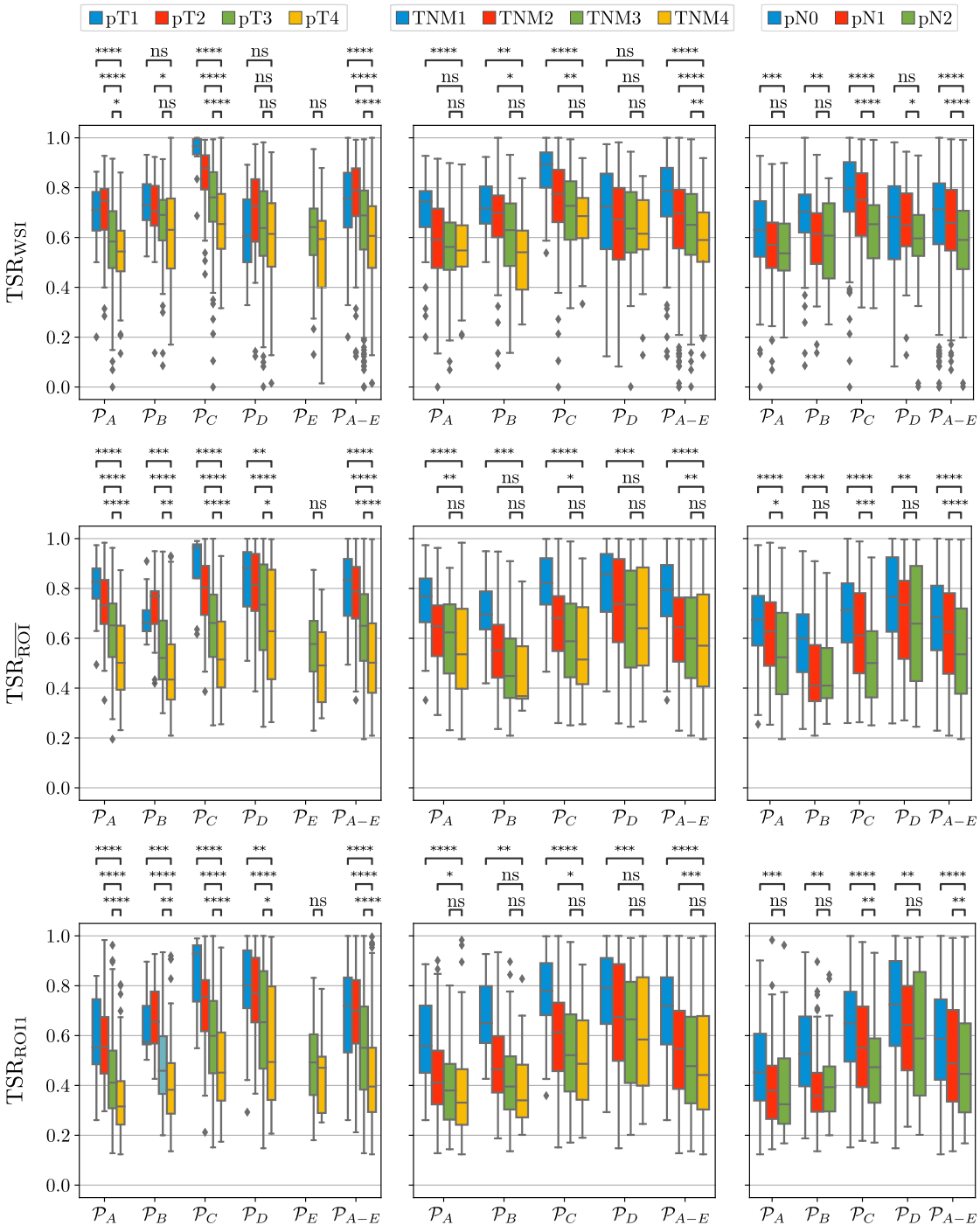
With the first case, we observe no variation of the TBC for all metrics and get a perfectly

pushing estimation of the overall TBC. In the second case, the "finger-like" growing pattern indicates an infiltrating border. All metrics can adequately detect the infiltrating part. The most interesting results come from the last case. Here, we see that the local estimation of TBC_{NP} varies a lot. As a result, we end up with regions estimated as pushing even though the pattern shows tumor budding. The remaining metrics are more stable. It comes from the fact the $\text{TBC}_{\text{RATIO}}$ and $\text{TBC}_{\text{INTER}}$ metric locally average the prediction, thus smoothing the prediction output. A solution to fix the estimation of TBC_{NP} would be to apply the same logic as for the other estimates and locally average the predictions for the normal vectors.

In Figure E.8, we display the estimation of the TB at the WSI-level. In Figure E.8a, we show the classification results over multiple classes along with the TB. For each point along the TB, the border is estimated as either pushing or infiltrating. In Figure E.8b, we present the main tissue areas (tumor, adipose, and muscle) used to orient the tissue. Finally, in Figure E.8c, we measure the local $\text{TBC}_{\text{INTER}}$ given a threshold δ_{INTER}.

E.6 Kaplan-Meier

We show the Kaplan-Meier (KM) estimates for overall survival (OS) and disease-free survival (DFS) in Figure E.9 and Figure E.10, respectively. The results are computed for the automated metrics TSR_{ROI1}, $\text{TBC}_{\text{INTER}}$, and TD_{STR} on all cohorts ($\mathcal{P}_{A-E}$). The metrics are selected based on their performance on the univariate model and stratified into two groups: low and high. For TSR and TBC, we use 50% as a cutoff. For the stroma distribution, the split is set to the group median (*i.e.* 0.2).



(a) Tumor depth (b) Stage (c) Lym. invasion

ns: $p \geq 0.05$, *: $p \leq 0.05$, **: $p \leq 0.01$, ***: $p \leq 0.001$, and ****: $p \leq 0.0001$

Figure E.6 – Correlation of the TSR feature with respect to other clinical values. We present the results for $\mathrm{TSR}_{\mathrm{WSI}}$, $\mathrm{TSR}_{\mathrm{ROI1}}$, $\mathrm{TSR}_{\overline{\mathrm{ROI}}}$ over five different cohort. The clinical features are as follows: (a) depth of invasion of the tumor, (b) the overall cancer stage, and (c) the number of lymph nodes affected. We indicate statistical relevance using the Mann–Whitney U test.

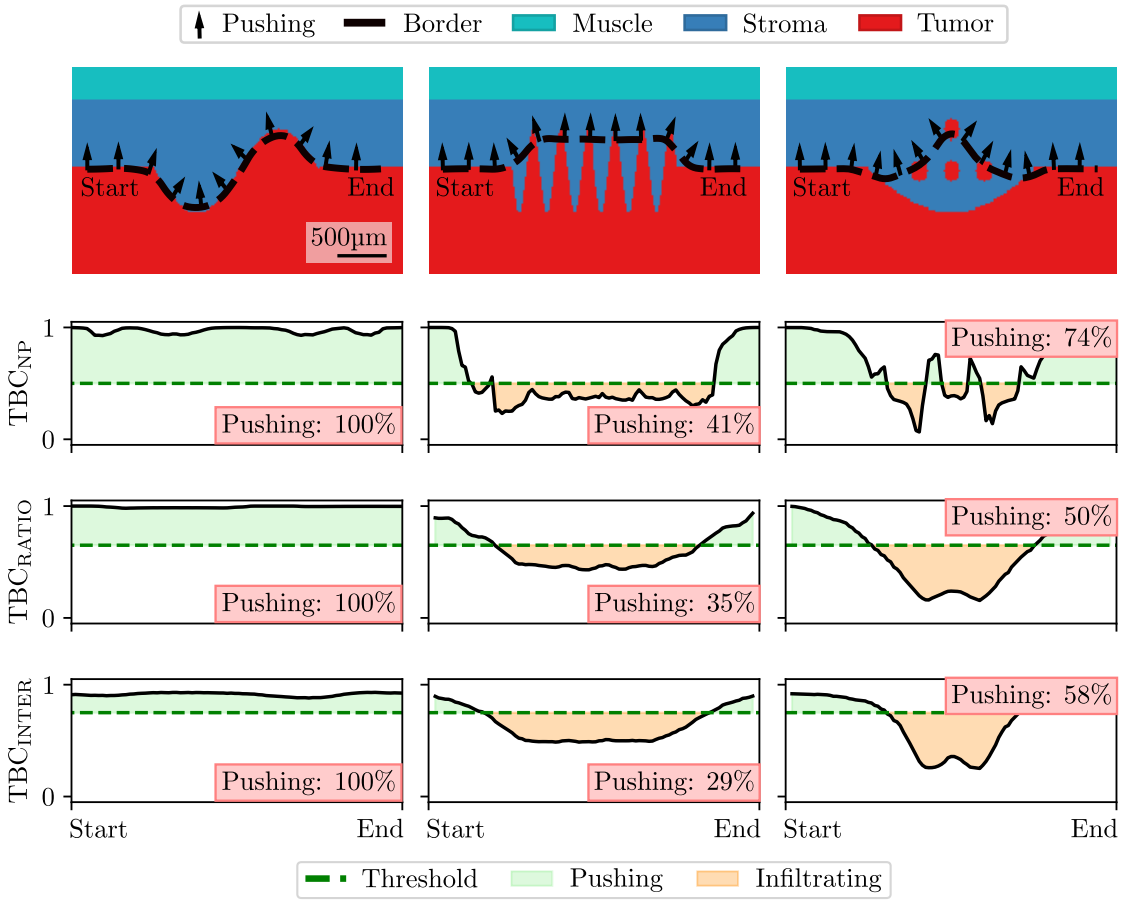
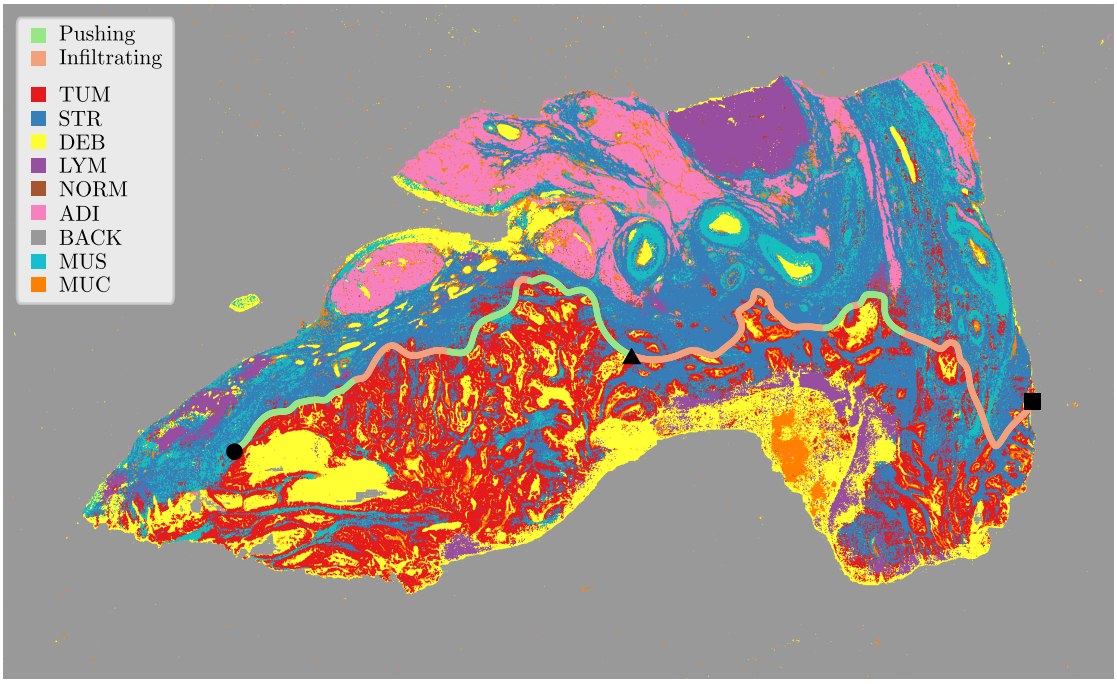
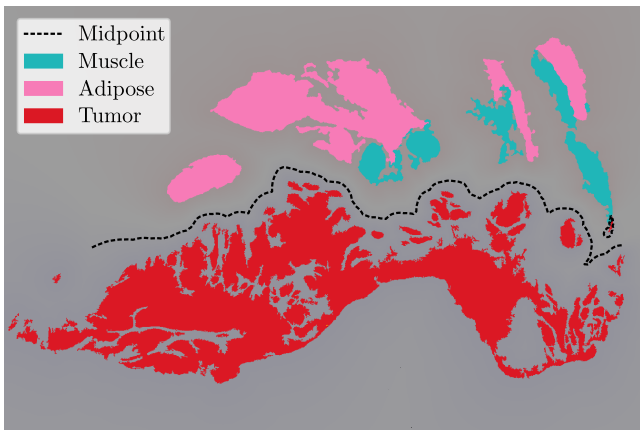


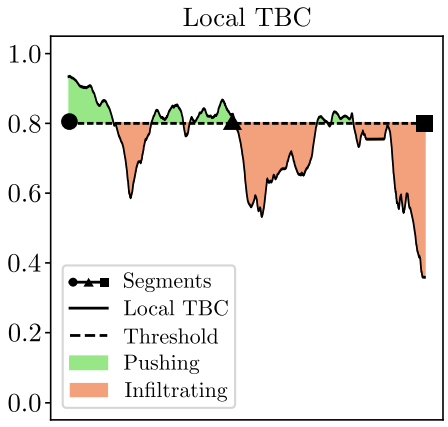
Figure E.7 – Toy examples for TBC estimation. Each column shows a different border configuration with tumor (red) progressing through stroma (blue), TB outline (dashed), and growing direction (arrows). The rows show the evolution of the TBC estimates along the tumor border (start to end). We use predefined thresholds for each metric.



(a) Segmentation of WSI and location of ROIs



(b) Decision boundary (healthy/tumor)



(c) Metric evolution

Figure E.8 – Visualization of TB and TBC estimation. (a) Segmentation of the WSI into multiple classes. For each point along the TB, we estimate if it is pushing or infiltrating. (b) Identification of main tissue blobs to build boundary. (c) Local TBC measure before thresholding for interaction (TBC_{INTER}).

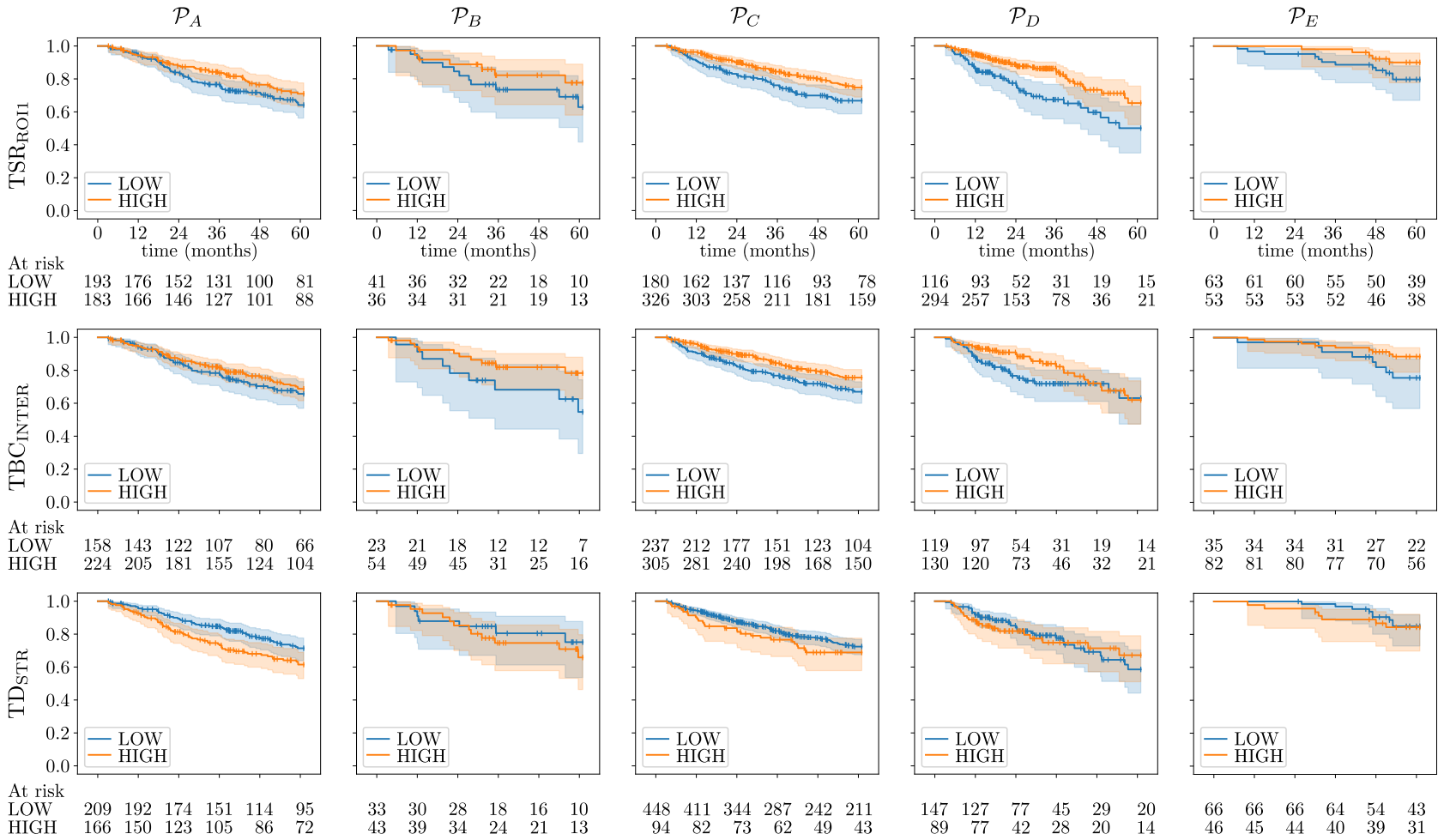


Figure E.9 – Overall survival and Kaplan-Meier estimates for TSR_{ROI1}, TBC_{INTER}, and TD_{STR} on all cohorts ($\mathcal{P}_{A-E}$). The metrics are stratified into low and high groups. The table below shows the number of at-risk patients at each time step.

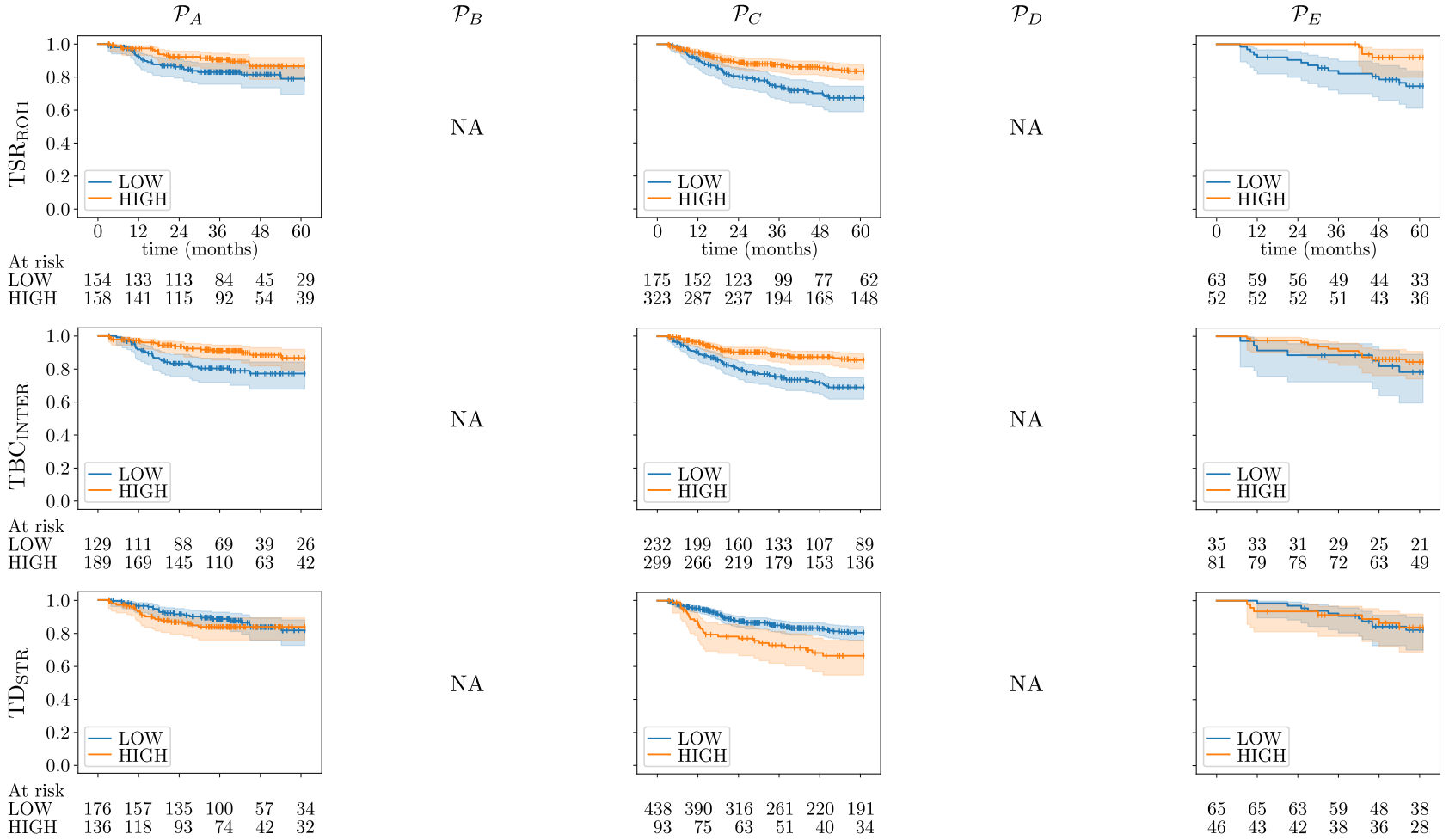


Figure E.10 – Disease-free survival and Kaplan-Meier estimates for TSR_{ROI1}, TBC_{INTER}, and TD_{STR} on all cohorts ($\mathcal{P}_{A-E}$). The metrics are stratified into low and high groups. The table below shows the number of at-risk patients at each time step. Not available (NA).

E.7 Proportional Hazards - Stage II

In Figure E.11 and Figure E.12, we report univariate and multivariate Cox proportional hazards (CPH) for stage II patients across all cohorts. We use the same settings as for the analysis over all CRC stage with 5-year study length and l^1-norm regularization ($\lambda = 0.01$). For the univariate case, we report results across the five different cohorts. Concerning the multivariate approach, we solely highlight the overall performance. We reduce the variables to the ones that are common to all cohorts.

Regarding the clinical variables for the univariate case, it is difficult to find a metric that improves patient stratification. For the OS, gender and lymphatic invasion highlight a difference across all cohorts. For DFS, only budding shows a distinction between the groups. In all cases, we cannot find a consistent metric for each cohort. When focusing on the TSR estimation, we observe good performances for the DFS prediction. Both TSR estimation based on ROIs and TA-STR shows statistical significance at the overall level. Surprisingly, we observe a clear difference with the estimation of TSR using TA-STR rather than the dummy WSI approach. The difference was not visible when we included all cancer stages. It validates the hypothesis that the stroma in the neighborhood of the main area contains relevant information for stage II patient stratification.

When focusing on the multivariate setting, specifically the OS case, we observe only a statistical significance of the clinical metric. Adding the results of the automated approaches does not improve the performance of the proportional hazards. For the DFS case, we notice major differences. First, none of the clinical metrics are selected by the model. In addition, only the automated TSR works for the overall estimation. Surprisingly, the aggregation of all metrics does not show significant results even though TSR has proven useful. This is because we work with a different subset of slides in the two settings. For the overall case, we consider entries where TSR_{ROI1}, $\text{TBC}_{\text{RATIO}}$ and TD_{STR} are available which is more restrictive.

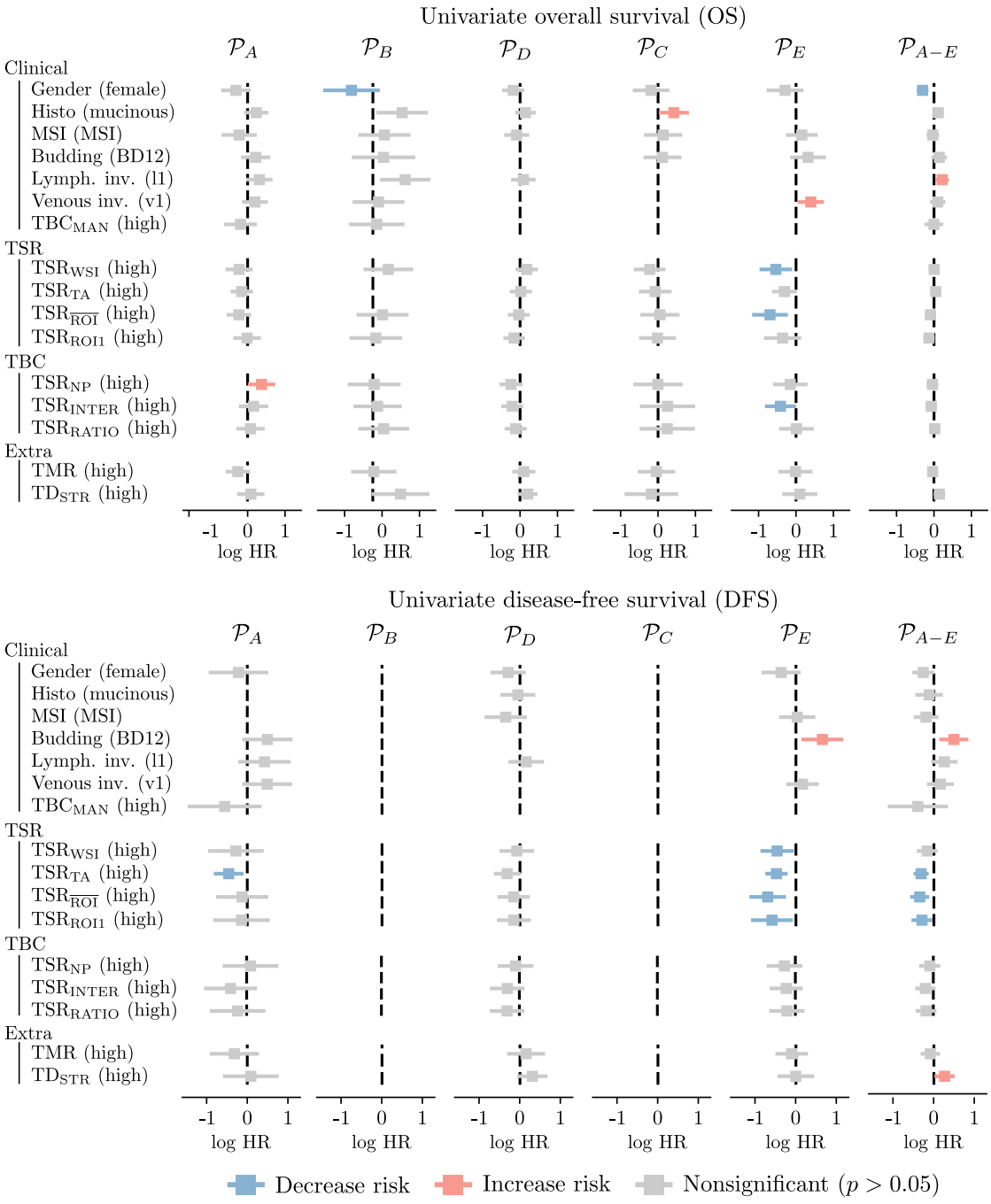


Figure E.11 – Univariate CPH estimation for OS and DFS based on clinical and automated metrics for stage II subset. We consider a 5-year period for the study where features are binarized before model fitting. The selected group is indicated between parentheses.

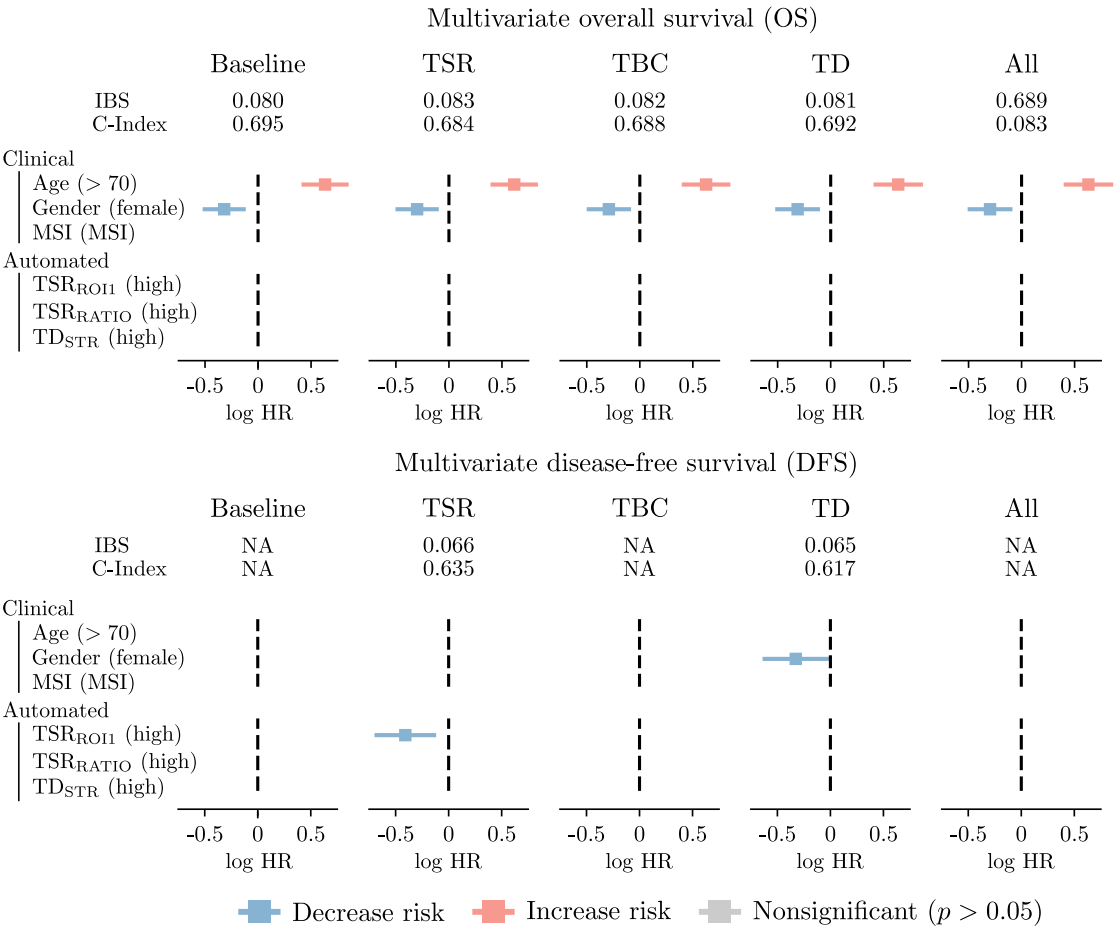


Figure E.12 – Multivariate CPH estimation for OS and DFS based on clinical and automated metrics for stage II subset. We consider a 5-year period for the study where features are binarized before model fitting. The selected group is indicated between parentheses. We report concordance index (C-Index) and integrated Brier score (IBS).

Bibliography

[1] C. Abbet, L. Studer, A. Fischer, H. Dawson, I. Zlobec, B. Bozorgtabar, and J.-P. Thiran. Self-rule to adapt: Learning generalized features from sparsely-labeled data using unsupervised domain adaptation for colorectal cancer tissue phenotyping. In *Medical Imaging with Deep Learning*, 2021.

[2] C. Abbet, L. Studer, A. Fischer, H. Dawson, I. Zlobec, B. Bozorgtabar, and J.-P. Thiran. Self-rule to multi-adapt: Generalized multi-source feature learning using unsupervised domain adaptation for colorectal cancer tissue detection. *Medical image analysis*, 79:102473, 2022.

[3] C. Abbet, L. Studer, I. Zlobec, and J.-P. Thiran. Toward automatic tumor-stroma ratio assessment for survival analysis in colorectal cancer. In *Medical Imaging with Deep Learning*, 2022.

[4] C. Abbet, I. Zlobec, B. Bozorgtabar, and J.-P. Thiran. Divide-and-rule: self-supervised learning for survival analysis in colorectal cancer. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference, Lima, Peru, October 4–8, 2020, Proceedings, Part V 23*, pages 480–489. Springer, 2020.

[5] Z. Allen-Zhu and Y. Li. Towards understanding ensemble, knowledge distillation and self-distillation in deep learning. *arXiv preprint arXiv:2012.09816*, 2020.

[6] M. B. Amin, S. B. Edge, F. L. Greene, D. R. Byrd, R. K. Brookland, M. K. Washington, J. E. Gershenwald, C. C. Compton, K. R. Hess, D. C. Sullivan, et al. *AJCC cancer staging manual*, volume 1024. Springer, 2017.

[7] D. Anand, G. Ramakrishnan, and A. Sethi. Fast gpu-enabled color normalization for digital pathology. In *2019 International Conference on Systems, Signals and Image Processing (IWSSIP)*, pages 219–224. IEEE, 2019.

[8] G. Aresta, T. Araújo, S. Kwok, S. S. Chennamsetty, M. Safwan, V. Alex, B. Marami, M. Prastawa, M. Chan, M. Donovan, et al. Bach: Grand challenge on breast cancer histology images. *Medical image analysis*, 56:122–139, 2019.

[9] M. Assran, M. Caron, I. Misra, P. Bojanowski, A. Joulin, N. Ballas, and M. Rabbat. Semi-supervised learning of visual features by non-parametrically predicting view assignments with support samples. *arXiv preprint arXiv:2104.13963*, 2021.

[10] R. Balestriero, M. Ibrahim, V. Sobal, A. Morcos, S. Shekhar, T. Goldstein, F. Bordes, A. Bardes, G. Mialon, Y. Tian, et al. A cookbook of self-supervised learning. *arXiv preprint arXiv:2304.12210*, 2023.

[11] P. Bankhead, M. B. Loughrey, J. A. Fernández, Y. Dombrowski, D. G. McArt, P. D. Dunne, S. McQuaid, R. T. Gray, L. J. Murray, H. G. Coleman, et al. Qupath: Open source software for digital pathology image analysis. *Scientific reports*, 7(1):1–7, 2017.

[12] H. Bao, L. Dong, S. Piao, and F. Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021.

[13] L. Berben, H. Wildiers, L. Marcelis, A. Antoranz, F. Bosisio, S. Hatse, and G. Floris. Computerised scoring protocol for identification and quantification of different immune cell populations in breast tumour regions by the use of qupath software. *Histopathology*, 77(1):79–91, 2020.

[14] M. Bilal, S. E. A. Raza, A. Azam, S. Graham, M. Ilyas, I. A. Cree, D. Snead, F. Minhas, and N. M. Rajpoot. Development and validation of a weakly supervised deep learning framework to predict the status of molecular pathways and key mutations in colorectal cancer from routine histology images: a retrospective study. *The Lancet Digital Health*, 3(12):e763–e772, 2021.

[15] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

[16] N. E. Breslow. Analysis of survival data under the proportional hazards model. *International Statistical Review/Revue Internationale de Statistique*, pages 45–57, 1975.

[17] G. W. Brier. Verification of forecasts expressed in terms of probability. *Monthly weather review*, 78(1):1–3, 1950.

[18] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[19] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, and A. A. Kalinin. Albumentations: Fast and flexible image augmentations. *Information*, 11(2), 2020.

[20] Cancer.Net Editorial Board. Colorectal cancer: Stages, 2022. https://www.cancer.net/cancer-types/colorectal-cancer [Accessed on June 21st, 2023].

[21] F. M. Carlucci, A. D'Innocente, S. Bucci, B. Caputo, and T. Tommasi. Domain generalization by solving jigsaw puzzles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2229–2238, 2019.

[22] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33:9912–9924, 2020.

[23] M. Caron, H. Touvron, I. Misra, H. Jégou, J. Mairal, P. Bojanowski, and A. Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.

[24] Center for disease control and prevention. Colorectal cancer: What are the risk factors?, 2022. https://www.cdc.gov/cancer/colorectal/basic_info/risk_factors.htm [Accessed on August 24th, 2023].

[25] L. Chan, M. S. Hosseini, C. Rowsell, K. N. Plataniotis, and S. Damaskinos. Histosegnet: Semantic segmentation of histological tissue type in whole slide images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 10662–10671, 2019.

[26] G. Chechik, V. Sharma, U. Shalit, and S. Bengio. Large scale online learning of image similarity through ranking. *Journal of Machine Learning Research*, 11(3), 2010.

[27] C. Chen, Q. Dou, H. Chen, J. Qin, and P. A. Heng. Unsupervised bidirectional cross-modality adaptation via deeply synergistic image and feature alignment for medical image segmentation. *IEEE Transactions on Medical Imaging*, 2020.

[28] J. Chen, W. Cranton, and M. Fihn. *Handbook of visual display technology*. Springer, 2016.

[29] R. J. Chen, C. Chen, Y. Li, T. Y. Chen, A. D. Trister, R. G. Krishnan, and F. Mahmood. Scaling vision transformers to gigapixel images via hierarchical self-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16144–16155, 2022.

[30] R. J. Chen and R. G. Krishnan. Self-supervised vision transformers learn visual concepts in histopathology. *arXiv preprint arXiv:2203.00585*, 2022.

[31] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[32] X. Chen, H. Fan, R. Girshick, and K. He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[33] X. Chen and K. He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15750–15758, 2021.

[34] X. Chen*, S. Xie*, and K. He. An empirical study of training self-supervised vision transformers. *arXiv preprint arXiv:2104.02057*, 2021.

[35] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 539–546. IEEE, 2005.

[36] O. Ciga, T. Xu, and A. L. Martel. Self supervised contrastive learning for digital histopathology. *Machine Learning with Applications*, 7:100198, 2022.

[37] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath. Generative adversarial networks: An overview. *IEEE signal processing magazine*, 35(1):53–65, 2018.

[38] M. Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.

[39] R. Damhuis, B. Wijnhoven, P. Plaisier, W. Kirkels, R. Kranse, and J. Van Lanschot. Comparison of 30-day, 90-day and in-hospital postoperative mortality for eight different cancer types. *Journal of British Surgery*, 99(8):1149–1154, 2012.

[40] C. Davidson-Pilon. lifelines: survival analysis in python. *Journal of Open Source Software*, 4(40):1317, 2019.

[41] B. Delaunay et al. Sur la sphere vide. *Izv. Akad. Nauk SSSR, Otdelenie Matematicheskii i Estestvennyka Nauk*, 7(793-800):1–2, 1934.

[42] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. IEEE, 2009.

[43] K. Dercksen, W. Bulten, and G. Litjens. Dealing with label scarcity in computational pathology: A use case in prostate cancer classification. *arXiv preprint arXiv:1905.06820*, 2019.

[44] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[45] P. Dierckx. *Curve and surface fitting with splines*. Oxford University Press, 1995.

[46] C. Doersch, A. Gupta, and A. A. Efros. Unsupervised visual representation learning by context prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1422–1430, 2015.

[47] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[48] Q. Dou, D. Coelho de Castro, K. Kamnitsas, and B. Glocker. Domain generalization via model-agnostic learning of semantic features. *Advances in Neural Information Processing Systems*, 32:6450–6461, 2019.

[49] B. Efron. The efficiency of cox's likelihood function for censored data. *Journal of the American statistical Association*, 72(359):557–565, 1977.

[50] K. Faryna, J. van der Laak, and G. Litjens. Tailoring automated data augmentation to h&e-stained histopathology. In *Medical Imaging with Deep Learning*, 2021.

[51] A. Filiot, R. Ghermi, A. Olivier, P. Jacob, L. Fidon, A. Mac Kain, C. Saillard, and J.-B. Schiratti. Scaling self-supervised learning for histopathology with masked image modeling. *medRxiv*, pages 2023–07, 2023.

[52] D. Firmbach, M. Benz, P. Kuritcyn, V. Bruns, C. Lang-Schwarz, F. A. Stuebs, S. Merkel, L.-S. Leikauf, A.-L. Braunschweig, A. Oldenburger, et al. Tumor-stroma ratio in colorectal cancer—comparison between human estimation and automated assessment. *Cancers*, 15(10):2675, 2023.

[53] S. Fouad, D. Randell, A. Galton, H. Mehanna, and G. Landini. Unsupervised morphological segmentation of tissue compartments in histopathological images. *PloS one*, 12(11):e0188717, 2017.

[54] Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189, 2015.

[55] Y. Ge, D. Chen, F. Zhu, R. Zhao, and H. Li. Self-paced contrastive learning with hybrid memory for domain adaptive object re-id. *arXiv preprint arXiv:2006.02713*, 2020.

[56] S. Gidaris, P. Singh, and N. Komodakis. Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728*, 2018.

[57] E. Graf, C. Schmoor, W. Sauerbrei, and M. Schumacher. Assessment and comparison of prognostic classification schemes for survival data. *Statistics in medicine*, 18(17-18):2529–2545, 1999.

[58] M. Greenwood. The natural duration of cancer (report on public health and medical subjects no 33). *London: Stationery Office*, 1926.

[59] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

[60] J. Guinney, R. Dienstmann, X. Wang, A. De Reynies, A. Schlicker, C. Soneson, L. Marisa, P. Roepman, G. Nyamundanda, P. Angelino, et al. The consensus molecular subtypes of colorectal cancer. *Nature medicine*, 21(11):1350–1356, 2015.

[61] T. S. Gurina and L. Simms. Histology, staining. 2023.

[62] M. Gutmann and A. Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 297–304. JMLR Workshop and Conference Proceedings, 2010.

[63] F. E. Harrell Jr, K. L. Lee, and D. B. Mark. Multivariable prognostic models: issues in developing models, evaluating assumptions and adequacy, and measuring and reducing errors. *Statistics in medicine*, 15(4):361–387, 1996.

[64] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

[65] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[66] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[67] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.

[68] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.

[69] Z. Huang, F. Bianchi, M. Yuksekgonul, T. Montine, and J. Zou. Leveraging medical twitter to build a visual–language foundation model for pathology ai. *bioRxiv*, pages 2023–03, 2023.

[70] J. Jass, S. Love, and J. Northover. A new prognostic classification of rectal cancer. *The Lancet*, 329(8545):1303–1306, 1987.

[71] S. Javed, A. Mahmood, M. M. Fraz, N. A. Koohbanani, K. Benes, Y.-W. Tsang, K. Hewitt, D. Epstein, D. Snead, and N. Rajpoot. Cellular community detection for tissue phenotyping in colorectal cancer histology images. *Medical Image Analysis*, page 101696, 2020.

[72] S. Jo and I.-J. Yu. Puzzle-cam: Improved localization via matching partial and full features. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 639–643. IEEE, 2021.

[73] S. Jo, I.-J. Yu, and K. Kim. Recurseed and edgepredictmix: Single-stage learning is sufficient for weakly-supervised semantic segmentation. *arXiv preprint arXiv:2204.06754*, 2022.

[74] M. L. Johnson, H. S. Gordon, N. J. Petersen, N. P. Wray, A. L. Shroyer, F. L. Grover, and J. M. Geraci. Effect of definition of mortality on hospital profiles. *Medical Care*, pages 7–16, 2002.

[75] M. Kang, H. Song, S. Park, D. Yoo, and S. Pereira. Benchmarking self-supervised learning on diverse pathology datasets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3344–3354, 2023.

[76] E. L. Kaplan and P. Meier. Nonparametric estimation from incomplete observations. *Journal of the American statistical association*, 53(282):457–481, 1958.

[77] J. N. Kather, N. Halama, and A. Marx. 100,000 histological images of human colorectal cancer and healthy tissue, Apr. 2018.

[78] J. N. Kather, J. Krisam, P. Charoentong, T. Luedde, E. Herpel, C.-A. Weis, T. Gaiser, A. Marx, N. A. Valous, D. Ferber, et al. Predicting survival from colorectal cancer histology slides using deep learning: A retrospective multicenter study. *PLoS medicine*, 16(1):e1002730, 2019.

[79] J. N. Kather, C.-A. Weis, F. Bianconi, S. M. Melchers, L. R. Schad, T. Gaiser, A. Marx, and F. G. Zöllner. Multi-class texture analysis in colorectal cancer histology. *Scientific reports*, 6:27988, 2016.

[80] A. Khan, A. Janowczyk, F. Müller, A. Blank, H. G. Nguyen, C. Abbet, L. Studer, A. Lugli, H. Dawson, J.-P. Thiran, et al. Impact of scanner variability on lymph node segmentation in computational pathology. *Journal of pathology informatics*, 13:100127, 2022.

[81] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan. Supervised contrastive learning. *arXiv preprint arXiv:2004.11362*, 2020.

[82] D. Kim, K. Saito, T.-H. Oh, B. A. Plummer, S. Sclaroff, and K. Saenko. Cross-domain self-supervised learning for domain adaptation with few source labels. *arXiv preprint arXiv:2003.08264*, 2020.

[83] S. Klein, M. Staring, K. Murphy, M. A. Viergever, and J. P. Pluim. Elastix: a toolbox for intensity-based medical image registration. *IEEE transactions on medical imaging*, 29(1):196–205, 2009.

[84] V. H. Koelzer and A. Lugli. The tumor border configuration of colorectal cancer as a histomorphological prognostic indicator. *Frontiers in oncology*, 4:29, 2014.

[85] N. A. Koohbanani, B. Unnikrishnan, S. A. Khurram, P. Krishnaswamy, and N. Rajpoot. Self-path: Self-supervision for classification of pathology images with limited annotations. *IEEE Transactions on Medical Imaging*, 2021.

[86] H. Kvamme, Ø. Borgan, and I. Scheel. Time-to-event prediction with neural networks and cox regression. *arXiv preprint arXiv:1907.00825*, 2019.

[87] G. Landini. Colour deconvolution, 2016. Accessed on April 19th, 2023.

[88] T. Lazard, M. Lerousseau, E. Decencière, and T. Walter. Giga-ssl: Self-supervised learning for gigapixel images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4304–4313, 2023.

[89] K. Li, H. Luo, L. Huang, H. Luo, and X. Zhu. Microsatellite instability: a review of what the oncologist should know. *Cancer cell international*, 20:1–13, 2020.

[90] R. Li, J. Yao, X. Zhu, Y. Li, and J. Huang. Graph cnn for survival analysis on whole slide pathological images. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 174–182. Springer, 2018.

[91] X. Li and R. Xu. *High-dimensional data analysis in cancer research*. Springer Science & Business Media, 2008.

[92] Ligue Suisse contre le cancer. Le cancer en suisse: les chiffres, 2022. https://www.liguecancer.ch/a-propos-du-cancer/les-chiffres-du-cancer/-dl-/fileadmin/downloads/sheets/chiffres-le-cancer-en-suisse.pdf [Accessed on June 21st, 2023].

[93] M. Loda, L. A. Mucci, M. L. Mittelstadt, M. Van Hemelrijck, and M. B. Cotter. *Pathology and epidemiology of cancer*. Springer, 2016.

[94] M. Y. Lu, B. Chen, D. F. Williamson, R. J. Chen, I. Liang, T. Ding, G. Jaume, I. Odintsov, A. Zhang, L. P. Le, et al. Towards a visual-language foundation model for computational pathology. *arXiv preprint arXiv:2307.12914*, 2023.

[95] A. Lugli, R. Kirsch, Y. Ajioka, F. Bosman, G. Cathomas, H. Dawson, H. El Zimaity, J.-F. Fléjou, T. P. Hansen, A. Hartmann, et al. Recommendations for reporting tumor budding in colorectal cancer based on the international tumor budding consensus conference (itbcc) 2016. *Modern pathology*, 30(9):1299–1311, 2017.

[96] M. Macenko, M. Niethammer, J. S. Marron, D. Borland, J. T. Woosley, Xiaojun Guan, C. Schmitt, and N. E. Thomas. A method for normalizing histology slides for quantitative analysis. In *2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, pages 1107–1110, June 2009.

[97] J. Mairal, F. Bach, J. Ponce, et al. Sparse modeling for image and vision processing. *Foundations and Trends® in Computer Graphics and Vision*, 8(2-3):85–283, 2014.

[98] O. Majek, A. Gondos, L. Jansen, K. Emrich, B. Holleczek, A. Katalinic, A. Nennecke, A. Eberle, H. Brenner, and G. C. S. W. Group. Sex differences in colorectal cancer survival: population-based analysis of 164,996 colorectal cancer patients in germany. *PloS one*, 8(7):e68077, 2013.

[99] M. L. McHugh. Interrater reliability: the kappa statistic. *Biochemia medica*, 22(3):276–282, 2012.

[100] L. McInnes, J. Healy, and J. Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[101] A. L. Mescher. *Junqueira's basic histology: text and atlas*. New York: McGraw Hill, 2018.

[102] O. S. Miettinen. Survival analysis: up from kaplan–meier–greenwood. *European journal of epidemiology*, 23(9):585–592, 2008.

[103] T. Moriya, H. R. Roth, S. Nakamura, H. Oda, K. Nagara, M. Oda, and K. Mori. Unsupervised pathology image segmentation using representation learning with spherical k-means. In *Medical Imaging 2018: Digital Pathology*, volume 10581, page 1058111. International Society for Optics and Photonics, 2018.

[104] H. Muhammad, C. S. Sigel, G. Campanella, T. Boerner, L. M. Pak, S. Büttner, J. N. IJzermans, B. G. Koerkamp, M. Doukas, W. R. Jarnagin, et al. Unsupervised subtyping of cholangiocarcinoma using a deep clustering convolutional autoencoder. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 604–612. Springer, 2019.

[105] MyPathologyReport.ca. Colon. what is the colon?, 2023.

[106] I. P. Nearchou, H. Ueno, Y. Kajiwara, K. Lillard, S. Mochizuki, K. Takeuchi, D. J. Harrison, and P. D. Caie. Automated detection and classification of desmoplastic reaction at the colorectal tumour front using deep learning. *Cancers*, 13(7):1615, 2021.

[107] H.-G. Nguyen, O. Lundström, A. Blank, H. Dawson, A. Lugli, M. Anisimova, and I. Zlobec. Image-based assessment of extracellular mucin-to-tumor area predicts consensus molecular subtypes (cms) in colorectal cancer. *Modern pathology*, 35(2):240–248, 2022.

[108] M. Noroozi and P. Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European Conference on Computer Vision*, pages 69–84. Springer, 2016.

[109] T. Okuyama, S. Sameshima, E. Takeshita, T. Mitsui, T. Noro, Y. Ono, T. Noie, S. Ban, and M. Oya. Myxoid stroma is associated with postoperative relapse in patients with stage ii colon cancer. *BMC cancer*, 20(1):1–11, 2020.

[110] A. v. d. Oord, Y. Li, and O. Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[111] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[112] X. Pan, Y. Gao, Z. Lin, F. Tang, W. Dong, H. Yuan, F. Huang, and C. Xu. Unveiling the potential of structure preserving for weakly supervised object localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11642–11651, 2021.

[113] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016.

[114] E. Pizzi, S. D. Roy, S. N. Ravindra, P. Goyal, and M. Douze. A self-supervised descriptor for image copy detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14532–14542, 2022.

[115] S. Pölsterl. scikit-survival: A library for time-to-event analysis built on top of scikit-learn. *Journal of Machine Learning Research*, 21(212):1–6, 2020.

[116] V. Prabhu, S. Khare, D. Kartik, and J. Hoffman. Sentry: Selective entropy optimization via committee consistency for unsupervised domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8558–8567, 2021.

[117] Y. Z. Qwaider, N. M. Sell, C. E. Stafford, H. Kunitake, J. C. Cusack, R. Ricciardi, L. G. Bordeianou, V. Deshpande, R. N. Goldstone, C. E. Cauley, et al. Infiltrating tumor border configuration is a poor prognostic factor in stage ii and iii colon adenocarcinoma. *Annals of Surgical Oncology*, 28:3408–3414, 2021.

[118] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[119] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pages 8821–8831. PMLR, 2021.

[120] N. Roland, G. Porter, B. Fish, and Z. Makura. Tumour assessment and staging: United kingdom national multidisciplinary guidelines. *The Journal of Laryngology & Otology*, 130(S2):S53–S58, 2016.

[121] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb: An efficient alternative to sift or surf. In *2011 International conference on computer vision*, pages 2564–2571. Ieee, 2011.

[122] A. C. Ruifrok, D. A. Johnston, et al. Quantification of histochemical staining by color deconvolution. *Analytical and quantitative cytology and histology*, 23(4):291–299, 2001.

[123] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada. Maximum classifier discrepancy for unsupervised domain adaptation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3723–3732, 2018.

[124] K. Saito, S. Yamamoto, Y. Ushiku, and T. Harada. Open set domain adaptation by backpropagation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 153–168, 2018.

[125] R. Sankaranarayanan, R. Swaminathan, H. Brenner, K. Chen, K. S. Chia, J. G. Chen, S. C. Law, Y.-O. Ahn, Y. B. Xiang, B. B. Yeole, et al. Cancer survival in africa, asia, and central america: a population-based study. *The lancet oncology*, 11(2):165–173, 2010.

[126] M. Schemper and T. L. Smith. A note on quantifying follow-up in studies of failure time. *Control clin trials*, 17:343–346, 1996.

[127] T. L. Sellaro, R. Filkins, C. Hoffman, J. L. Fine, J. Ho, A. V. Parwani, L. Pantanowitz, and M. Montalto. Relationship between magnification and resolution in digital pathology systems. *Journal of pathology informatics*, 4(1):21, 2013.

[128] K. Shanah, S. Cheryl A., S. J. Keith, L. Seth, R. Charles, B. Ermalinda, and F. Joe. Radiology data from the cancer genome atlas colon adenocarcinoma [tcga-coad] collection, 2016.

[129] K. Shanah, S. Cheryl A., and L. Seth. Radiology data from the cancer genome atlas rectum adenocarcinoma [tcga-read] collection, 2016.

[130] S. Shurrab and R. Duwairi. Self-supervised learning methods and applications in medical imaging analysis: A survey. *PeerJ Computer Science*, 8:e1045, 2022.

[131] R. M. Souza da Silva, E. M. Queiroga, A. R. Paz, F. F. Neves, K. S. Cunha, and E. P. Dias. Standardized assessment of the tumor-stroma ratio in colorectal cancer: interobserver validation and reproducibility of a potential prognostic factor. *Clinical pathology*, 14:2632010X21989686, 2021.

[132] T. Stegmüller, C. Abbet, B. Bozorgtabar, H. Clarke, P. Petignat, P. Vassilakos, and J.-P. Thiran. Self-supervised learning-based cervical cytology diagnostics in low-data regime and low-resource setting. *arXiv preprint arXiv:2302.05195*, 2023.

[133] L. Sullivan, R. R. Pacheco, M. Kmeid, A. Chen, and H. Lee. Tumor stroma ratio and its significance in locally advanced colorectal cancer. *Current Oncology*, 29(5):3232–3241, 2022.

[134] A. Tam, J. Barker, and D. Rubin. A method for normalizing pathology images to improve feature extraction for quantitative pathology. *Medical physics*, 43(1):528–537, 2016.

[135] D. Tellez, M. Balkenhol, I. Otte-Höller, R. van de Loo, R. Vogels, P. Bult, C. Wauters, W. Vreuls, S. Mol, N. Karssemeijer, et al. Whole-slide mitosis detection in h&e breast histology using phh3 as a reference to train distilled stain-invariant convolutional networks. *IEEE transactions on medical imaging*, 37(9):2126–2136, 2018.

[136] D. Tellez, G. Litjens, P. Bándi, W. Bulten, J.-M. Bokhorst, F. Ciompi, and J. van der Laak. Quantifying the effects of data augmentation and stain color normalization in convolutional neural networks for computational pathology. *Medical image analysis*, 58:101544, 2019.

[137] D. Tellez, J. van der Laak, and F. Ciompi. Gigapixel whole-slide image classification using unsupervised image compression and contrastive training. *Medical Imaging with Deep Learning*, 2018.

[138] H. Ueno, Y. Kajiwara, Y. Ajioka, T. Sugai, S. Sekine, M. Ishiguro, A. Takashima, and Y. Kanemitsu. Histopathological atlas of desmoplastic reaction characterization in colorectal cancer. *Japanese Journal of Clinical Oncology*, 51(6):1004–1012, 2021.

[139] A. Vahadane, T. Peng, A. Sethi, S. Albarqouni, L. Wang, M. Baust, K. Steiger, A. M. Schlitter, I. Esposito, and N. Navab. Structure-preserving color normalization and sparse stain separation for histological images. *IEEE Transactions on Medical Imaging*, 35(8):1962–1971, Aug 2016.

[140] L. Van der Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[141] G. W. van Pelt, T. P. Sandberg, H. Morreau, H. Gelderblom, J. H. J. van Krieken, R. A. Tollenaar, and W. E. Mesker. The tumour–stroma ratio in colon cancer: the biological role and its prognostic impact. *Histopathology*, 73(2):197–206, 2018.

[142] B. S. Veeling, J. Linmans, J. Winkens, T. Cohen, and M. Welling. Rotation equivariant cnns for digital pathology. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2018: 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part II 11*, pages 210–218. Springer, 2018.

[143] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, P.-A. Manzagol, and L. Bottou. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(12), 2010.

[144] R. Viñals and J.-P. Thiran. A KL Divergence-Based Loss for In Vivo Ultrafast Ultrasound Image Enhancement with Deep Learning. 8 2023.

[145] F. Wang and H. Liu. Understanding the behaviour of contrastive loss. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2495–2504, 2021.

[146] X. Wang, S. Yang, J. Zhang, M. Wang, J. Zhang, J. Huang, W. Yang, and X. Han. Transpath: Transformer-based self-supervised learning for histopathological image classification. In *Medical Image Computing and Computer Assisted Intervention–MICCAI 2021: 24th International Conference, Strasbourg, France, September 27–October 1, 2021, Proceedings, Part VIII 24*, pages 186–195. Springer, 2021.

[147] X. Wang, S. Yang, J. Zhang, M. Wang, J. Zhang, W. Yang, J. Huang, and X. Han. Transformer-based unsupervised contrastive learning for histopathological image classification. *Medical image analysis*, 81:102559, 2022.

[148] J. Wei, A. Suriawinata, B. Ren, X. Liu, M. Lisovsky, L. Vaickus, C. Brown, M. Baker, N. Tomita, L. Torresani, et al. A petri dish for histopathology image analysis. In *Artificial Intelligence in Medicine: 19th International Conference on Artificial Intelligence in Medicine, AIME 2021, Virtual Event, June 15–18, 2021, Proceedings*, pages 11–24. Springer, 2021.

[149] A. White, L. Ironmonger, R. J. Steele, N. Ormiston-Smith, C. Crawford, and A. Seims. A review of sex-related differences in colorectal cancer incidence, screening uptake, routes to diagnosis, cancer stage and survival in the uk. *BMC cancer*, 18(1):1–11, 2018.

[150] T. Wu, L. Pan, J. Zhang, T. Wang, Z. Liu, and D. Lin. Density-aware chamfer distance as a comprehensive metric for point cloud completion. *arXiv preprint arXiv:2111.12702*, 2021.

[151] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3733–3742, 2018.

[152] J. Xie, R. Girshick, and A. Farhadi. Unsupervised deep embedding for clustering analysis. In *International conference on machine learning*, pages 478–487, 2016.

[153] J. Xu, L. Xiao, and A. M. López. Self-supervised domain adaptation for computer vision tasks. *IEEE Access*, 7:156694–156706, 2019.

[154] Q. Xu, Y.-Y. Chen, Y.-H. Luo, J.-S. Zheng, Z.-H. Lin, B. Xiong, and L.-W. Wang. Proposal of an automated tumor-stromal ratio assessment algorithm and a nomogram for prognosis in early-stage invasive breast cancer. *Cancer Medicine*, 12(1):131–145, 2023.

[155] P. Yang, X. Yin, H. Lu, Z. Hu, X. Zhang, R. Jiang, and H. Lv. Cs-co: A hybrid self-supervised visual representation learning method for h&e-stained histopathological images. *Medical Image Analysis*, 81:102539, 2022.

[156] F. G. Zanjani, S. Zinger, et al. Deep convolutional gaussian mixture model for stain-color normalization of histopathological images. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 274–282. Springer, 2018.

[157] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny. Barlow twins: Self-supervised learning via redundancy reduction. In *International Conference on Machine Learning*, pages 12310–12320. PMLR, 2021.

[158] R. Zhang, P. Isola, and A. A. Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1058–1067, 2017.

[159] K. Zhao, Z. Li, S. Yao, Y. Wang, X. Wu, Z. Xu, L. Wu, Y. Huang, C. Liang, and Z. Liu. Artificial intelligence quantified tumour-stroma ratio is an independent predictor for overall survival in resectable colorectal cancer. *EBioMedicine*, 61:103054, 2020.

[160] S. Zheng, J. Lu, H. Zhao, X. Zhu, Z. Luo, Y. Wang, Y. Fu, J. Feng, T. Xiang, P. H. Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6881–6890, 2021.

[161] S. Zhong. Efficient online spherical k-means clustering. In *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, volume 5, pages 3180–3185. IEEE, 2005.

[162] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016.

[163] J. Zhou, C. Wei, H. Wang, W. Shen, C. Xie, A. Yuille, and T. Kong. ibot: Image bert pre-training with online tokenizer. *arXiv preprint arXiv:2111.07832*, 2021.

[164] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.

[165] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.

[166] X. Zhu, J. Yao, F. Zhu, and J. Huang. Wsisa: Making survival prediction from whole slide histopathological images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7234–7242, 2017.

[167] C. Zhuang, A. L. Zhai, and D. Yamins. Local aggregation for unsupervised learning of visual embeddings. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6002–6012, 2019.

[168] I. Zlobec, K. Baker, P. Minoo, S. Hayashi, L. Terracciano, and A. Lugli. Tumor border configuration added to tnm staging better stratifies stage ii colorectal cancer patients into prognostic subgroups. *Cancer: Interdisciplinary International Journal of the American Cancer Society*, 115(17):4021–4029, 2009.

Christian Abbet

abbet.christian@gmail.com christianabbet Christian Abbet

Employment History

2019 – 2023	**PhD Thesis** LTS5, EPFL, Lausanne.
2018 – 2019	**Research Assistant** LTS5, EPFL, Lausanne.
2018	**Master Thesis** Swisscom Digital Lab, Lausanne.
2015 – 2016	**Internship** Idiap Research Institute, Martigny.
2015	**Summer Internship** Technis SA, Lausanne.
2010-2011	**Internship** IRO Institut, Sion.

Education

2016 – 2018	**Master Degree in Information Technology** EPFL, Lausanne.
2012 – 2015	**Bachelor Degree in Electrical Engineering** EPFL, Lausanne.
2011 – 2012	**Preparatory year (CMS)** EPFL, Lausanne.
2007 – 2011	**Federal Vocational Baccalaureate (FVB)** Centre de Formation Professionnelle, Sion.
	Federal Vocational Education and Training (VET) École des Métiers du Valais, Sion.

Research Publications

Journal Articles

1. A. Bugnon, R. Viñals, **C. Abbet**, *et al.*, "Apiculture–des réseaux de neurones artificiels pour lutter contre le varroa,"
2. A. L. Frei, R. Oberson, E. Baumann, *et al.*, "Pathologist computer-aided diagnostic scoring of tumor cell fraction: A swiss national study," *Modern Pathology*, p. 100 335, 2023.
3. **C. Abbet**, L. Studer, A. Fischer, *et al.*, "Self-rule to multi-adapt: Generalized multi-source feature learning using unsupervised domain adaptation for colorectal cancer tissue detection," *Medical image analysis*, vol. 79, p. 102 473, 2022.
4. A. Khan, A. Janowczyk, F. Müller, *et al.*, "Impact of scanner variability on lymph node segmentation in computational pathology," *Journal of pathology informatics*, vol. 13, p. 100 127, 2022.

Conference Proceedings

1. **C. Abbet**, L. Studer, I. Zlobec, and J.-P. Thiran, "Toward automatic tumor-stroma ratio assessment for survival analysis in colorectal cancer," in *Medical Imaging with Deep Learning*, 2022.
2. **C. Abbet**, L. Studer, A. Fischer, *et al.*, "Self-rule to adapt: Learning generalized features from sparsely-labeled data using unsupervised domain adaptation for colorectal cancer tissue phenotyping," in *Medical Imaging with Deep Learning*, 2021.
3. **C. Abbet**, I. Zlobec, B. Bozorgtabar, and J.-P. Thiran, "Divide-and-rule: Self-supervised learning for survival analysis in colorectal cancer," in *Medical Image Computing and Computer Assisted Intervention–MICCAI 2020: 23rd International Conference*, Springer, 2020, pp. 480–489.

4. **C. Abbet**, M. M'hamdi, A. Giannakopoulos, *et al.*, "Churn intent detection in multilingual chatbot conversations and social media," in *Proceedings of the 22nd Conference on Computational Natural Language Learning*, 2018, pp. 161–170.

Unpublished / In preparation

1. T. Stegmüller, **C. Abbet**, B. Bozorgtabar, *et al.*, "Self-supervised learning-based cervical cytology diagnostics in low-data regime and low-resource setting," 2023.

Abstracts

1. **C. Abbet**, L. Studer, J.-P. Thiran, and I. Zlobec, *Impact of scanner variability on colorectal cancer tumor segmentation*, 19th European congress on digital pathology – ECDP, 2023.
2. **C. Abbet**, L. Studer, J.-P. Thiran, and I. Zlobec, *Self-rule to multi adapt automates the tumor-stroma assessment in colorectal cancer*, 18th European congress on digital pathology – ECDP, 2022.
3. B. Elias, **C. Abbet**, and I. Zlobec, *Automatic quantification of "myxoid" desmoplastic stroma in colorectal cancer: A heterogeneous feature and challenging task*, 18th European congress on digital pathology – ECDP, 2022.
4. **C. Abbet**, L. Studer, J.-P. Thiran, and I. Zlobec, *Reducing the annotation workload: Using self-supervised methods to learn from publicly available colorectal cancer datasets*, 87th Annual Congress SSPath and 2nd Swiss Pathology Days, 2021.
5. A. Frei, **C. Abbet**, and I. Zlobec, *Cell-based graphs for tissue classification in colorectal histopathology images*, 87th Annual Congress SSPath and 2nd Swiss Pathology Days, 2021.
6. **C. Abbet**, L. Studer, J.-P. Thiran, and I. Zlobec, *Unsupervised joint clustering and representation learning for survival analysis in colorectal cancer*, 16th European congress on digital pathology – ECDP, 2020.

Languages

French	Mother tongue
Italian	Strong reading and speaking competencies
English	Strong reading, writing and speaking competencies (C1)
German	School level

Miscellaneous Experience

Awards and Achievements

2023	**Talk**. Abstract accepted for talk at the European Congress on Digital Pathology.
2022	**STI Teaching Assistant Award**. Awarded by the School of Engineering of EPFL.
	Talk. Abstract accepted for talk at Swiss Pathology Days.
2020	**Online Talk**. Conference paper accepted for talk at the International Conference on Medical Image.
2017	**Open Food Hackdays winner**. Awarded by Open Food.
2016	**Winner of Logitech challenge**. Awarded by Lauzhack.

Certifications

2018	**C1 Advanced**. Awarded by Cambridge English.